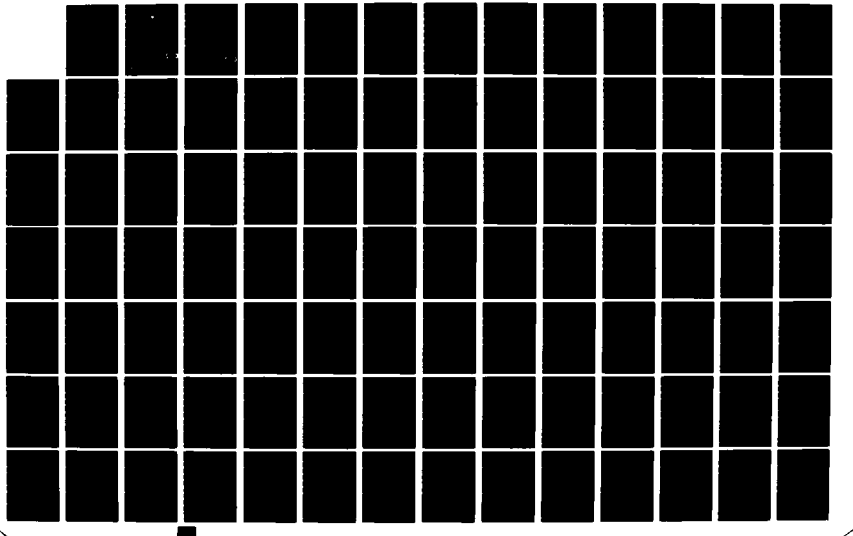
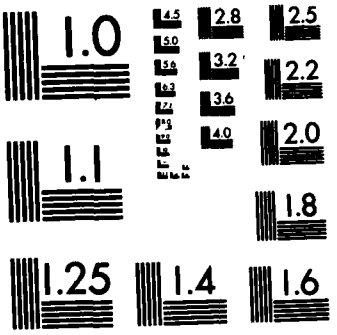
AD-A124 820 A SECURE COMPUTER NETWORK(U) AIR FORCE INST OF TECH 1/2
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
J S STEINMETZ NOV 82 AFIT/GCS/EE/82D-34

UNCLASSIFIED F/G 9/2 NL





A SECURE COMPUTER NETWORK

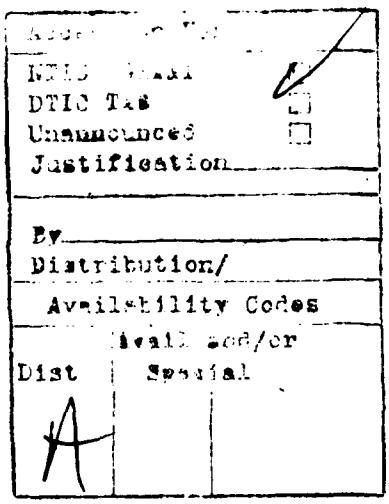
THESIS

AFIT/GCS/EE/82D-34 Jay S. Steinmetz
Captain USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

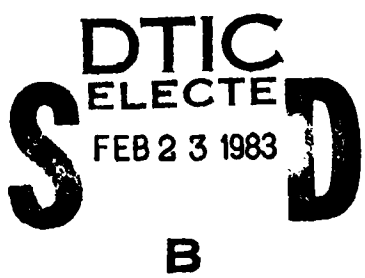
AFIT/GCS/EE/82D-34



A SECURE COMPUTER NETWORK

THESIS

AFIT/GCS/EE/82D-34 Jay S. Steinmetz
Captain USAF



B

Approved for public release; distribution unlimited.

AFIT/GCS/EE/82D-34

A SECURE COMPUTER NETWORK

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Jay S. Steinmetz, B.S.
Captain USAF
Graduate Computer Systems
November 1982

Preface

Computer networks must have the capability to protect the information they contain. A network which does not provide secure processing and storage for information either cannot be used to process sensitive or personal information, or will compromise the security of that information. While many applications exist for a network capable of secure information processing, no such network is currently available. This research attempts to initiate the design and certification of a network capable of providing information security.

I thank my thesis advisor, Major Walter D. Seward, for his careful consideration of the many thoughts I have presented to him. His patience, thoughtful questions, and willingness for me to direct my own research made this thesis effort a rewarding experience. I also thank the other members of my thesis committee, Dr. Thomas C. Hartrum and First Lieutenant Robert W. Milne, for their probing questions about my design and their close reviews of my thesis drafts. Finally, I thank my wife, Luellen, for her love and support throughout the completion of this research.

Jay S. Steinmetz

Contents

Contents

List of Figures

List of Tables

Abstract

In this thesis, the initial design for a secure computer network is developed. The requirement for a secure computer network is based on the need to protect sensitive and personal information currently processed by computer networks. The concepts of physical security, reference monitors, encryption, and network protocols are presented. Then, the top-level design of the secure computer network is developed. This design consists of secure data bases controlled by kernelized secure operating systems which are connected by a secure communications subnetwork. The phases of secure communications: location, identification, request, and request response are discussed. A model for the secure communications subnetwork is then presented. This model relies on two major components: Secure Network Interface Computers (SNICs) and a Network Directory and Security Center (NDSC). A finite state analysis of the communication channels demonstrates the security of the model. Recommendations are presented to continue the development of this secure network.

A SECURE COMPUTER NETWORK

I Introduction

Computers and networks of computers have tremendously increased man's ability to gather, process, store, and analyze vast amounts of information. The incredible growth of computer facilities and availability of information has improved communications and enabled millions of people to stay informed of the latest developments and the current status of almost every aspect of our daily lives. Making all of this computerized information available to anyone desiring to use it may have serious repercussions. Not only are there legal ramnifications for failing to protect personal information (Turn, 1976), but there may also be costly consequenses for failing to protect information which is vital to the successful operation of an organization (Parker, 1976). Attempts have been made to apply security controls for the information stored in and processed by the computer networks spanning our country, but a reliable, secure computer network, which permits every user to process information securely, concurrently with other users, has yet to be developed. This thesis provides the initial design of such a network and develops a model which provides complete information security for individual

network users.

Computer Networks

A computer network is a group of computers with the ability to communicate with each other. The network includes any and all resources of every computer system connected to the network. These computers may reside in the same building or be spread thousands of miles apart. They may all be the same model of computer, or each computer may be a different type. The network may link supercomputers, minicomputers, and microcomputers. The network may operate through one or more centralized computer systems, or may distribute the network functions to every computer involved in the network. The network configuration may even change dynamically. The variations of networks which are possible are infinite. The monetary, computational, and temporal expenditures to create and maintain these networks can be quite considerable. These networks are created and maintained, however, because they provide several advantages over independent computer systems.

Networks allow access to computer resources (including the information contained in the network's computer systems) which are not, or cannot be, maintained in the local systems. This prevents unnecessary duplication of data bases and other computer resources, and still allows local control of local resources. (Local control, however, is not always maintained.) Networks

also allow each of their users to gain access (potentially) to any of the information or other computer resources contained in the network. This last advantage has created a tremendous problem.

Information Security

Information may be considered secure when only the people or machines with the proper authorization are allowed receive, copy, modify, or destroy that information. Of course, this implies that the people and machines authorized to use the information are trustworthy. It will be assumed that individuals authorized to use information are trustworthy. The determination of authorization and the prevention of unauthorized information use still remain complex tasks. The automation of information processing has been implemented much more quickly than have adequate safeguards to protect the information, either while it is stored in the system or processed by the system. This lack of protection has compromised the security of tremendous amounts of personal and sensitive information (Parker, 1976). The creation of computer networks, which link many completely unsecured or ineffectively-secured computer facilities together, has compounded the information security problem (Shen, 1974: 21).

Information is a valuable resource and should be protected accordingly. No longer is it just a nicety to protect information. Adequate information security is a

necessity. The legal, moral, and economic implications of the failure to adequately protect the information processed by computer systems and computer networks are staggering. The damage which could be caused by copying, modifying, deleting, or adding information to any data base or information transmittal is overwhelming. Consider the implications of illegal changes to a bank's records, a company's personnel file, a company's inventory records, or simply a company's mailing lists. Many of these data bases are changed every day (legally and illegally) through the use of computer networks which are not secure. Very few network applications can tolerate illegal changes to, or illegal copies of, the information they process.

Security procedures and policies should always be in effect if the information in the network is to be protected properly. A computer network should require no user direction to protect the information it already contains, but should require users to identify the security level of new information they introduce into the network. Therefore, the security measures should be automatically invoked by the network for all transactions, without specific user direction. In fact, if users are not introducing new information into the network, they do not need to be aware that security controls are in effect. While some identification process will be required for all network users, additional interactions for security should not be required unless a potential compromise is detected.

Attempts to secure computer systems or networks which were not designed for security have consistantly failed (Anderson, 1972). This failure increases the need for a properly designed computer network which can provide adequate security for the information it processes.

The tremendous capability of computers to handle large amounts of information makes them an attractive target for information thieves. Computer resource managers, therefore, have an inherent responsibility to protect the information processed by their computer facilities, and to allow access to the information with which they are entrusted only in accordance with the wishes of the "owners" of the information.

Objective

The objective of this thesis is to develop a computer network which maintains security for the information it contains or processes.

Approach and Scope

A secure system may only be built after first defining a conceptual design which provides the required security (Anderson, 1972: 11). This thesis, therefore, presents the design for a "complete" secure computer network model, rather than attempting to secure an existing network. A top-down, modular approach, in which information security is the dominant consideration, is followed in this research. The top-level design consists of three major components: secure data bases, secure

operating systems, and secure communications between the network computer systems. Only the secure communications between the network components will be developed below the top level. Current research and development in data base security and secure operating systems should be able to provide the other components necessary to complete the secure network suggested in this paper. Work on the UCLA Secure Unix Operating System (Popek, 1979) and the KSOS concept (McCauley, 1979; Berson, 1979; Padlipsky, 1979) is continuing with substantial gains in developing secure operating systems. Work is also progressing on the development of secure data bases (Davies, 1981; Denning, 1979; Hsiao, 1979; Turn, 1981: 163-214).

The top-down approach used in this thesis is important because it requires the development of the entire network - the total information environment. The "complete" network, including all of the components of each system on the network, must be considered if the information is to be properly protected. If the data bases or the operating systems are not secure, then the information handled by those components is not secure, and the value of the secure computer communications being developed here is extremely limited for protecting the information. To secure the communication medium, without securing the computer systems sending or recieving the information would be the same as transporting a bag of gold in an armored car and then delivering the bag to an

unlocked, unguarded vault. This does not suggest that the security for the armored car and the destination vault can not be developed separately, but rather that they must both be developed if the gold is to be properly protected. The information stored, processed, and transported in a computer network must be protected throughout the network if it is to be considered secure.

Organization

This thesis is presented so that each chapter represents a major phase of research. Though an attempt is made to make each chapter as independent as possible, the information presented in earlier chapters may be required, to fully understand the material. Concepts necessary to understand the design of the secure computer network developed in this paper are introduced in Chapter II. These include physical security, reference monitors, encryption, and network protocols. This introduction is not intended to be exhaustive or complete (because that would require volumes of information), but rather is intended to provide a basic understanding of the concepts used in this network model. The top-level secure network model, consisting of secure data bases, secure operating systems, and secure communications between the network components, is presented in Chapter III. Only the secure communications module of the model is developed past the top-level. Chapters IV presents the phases of secure communication. Chapter V more fully describes the secure

communications module by identifying its components, by describing the method it uses to accomplish the secure communication phases, and by listing the specific functions which must be performed by each component of the secure communications network. The model is analyzed and its securtiy is demonstrated in Chapter VI. Chapter VII presents the conclusions of this research and the recommendations for continued work.

Summary

Computer networks, which process vast amounts of personal or sensitive information, are not adequately protecting the information they contain because information security was not an initial design consideration for the networks. This thesis presents the top-level design of a secure computer network, and develops one of the three major modules of that design - the communications network. This communications network is a "complete" module which receives information, transports it, and delivers it to the proper destination. A complete, secure computer network could be constructed by adding secure computer systems, containing secure data bases, to the communications network developed here. Further refinement of this model will be necessary for physical implementation. This thesis presents a generalized, secure computer network model, which could be implemented for a variety of applications and conditions.

II Concepts

This chapter presents several concepts required to understand a secure computer communication network model. Physical security, reference monitors, encryption, and network protocols will be discussed. This discussion is not intended to provide a comprehensive understanding of these subjects, but rather to introduce the fundamentals of each subject and to show its relevence to a secure network model. Sources which more fully describe the topics introduced here are also identified.

Physical Security

Perhaps the oldest method of protecting valuable resources is the use of physical security. By carefully controlling the physical access to or exposure of specific resources, we can prevent (or significantly reduce the possibility of) loss or destruction of those resources.

Complete physical security can prevent unauthorized access to the information in a local computer system. If access to all of the system's components are positively controlled and only individuals authorized to access specific information are allowed to use the system components when that information is present in the system, then the information can be protected. However, this rather costly and archaic method of protecting information, still used in many systems today, precludes the connection of such a system to a network. Physical

security, therefore, is not sufficient (in the traditional sense) to prevent the compromise or destruction of information contained in or processed by a computer network. Simply guarding all of the components may reduce the chance of compromise, but will not, in a multi-user or multilevel computer network, prevent unauthorized disclosure, modification, addition, or destruction of information, since anyone with access to the network may have access to the information contained in the network. If, however, each piece of information is considered a resource, and access to the resources is carefully controlled, the traditional concept of physical security can be extended and applied in an environment where "physical" access includes "electronic" access.

Information security may be viewed as a set of access control barriers, or checkpoints, which must be sucessfully negotiated in order to gain access to the information contained in the network. If the only possible way to access the information is through the set of security controls located at each checkpoint, and the controls properly restrict access, then the information is secure. These access control barriers may be employed to prevent unauthorized access to the computer site, to system terminals, to files maintained by the system, and to data within the files.

Physical security traditionally includes measures to protect information from loss or destruction due to fire,

flood, or similar disasters. It also includes measures to prevent the information from leaving the computer site without the proper authorization, either in the form of trash or electromagnetic radiation, or while stored on tapes, paper, or disks. While procedures should be established to prevent such losses, these areas will not be discussed further in this paper. Physical security is discussed and additional references are identified in Computer Security (Hsiao, 1979).

If every component in a computer network could be physically secured, and access to each component (including every piece of information) could be carefully controlled, there would no longer be an information security problem. However, complete physical security is just not possible in most distributed computer networks. Physically securing the communications lines connecting geographically distributed components is extremely difficult. Many of these connections may be leased telephone lines or satellite links. Some method other than physical security must be employed to protect the information transmitted over these lines. Without proper protection, it would be a simple task for someone to "eavesdrop" on a communication or to masquerade as a legitimate network user by patching in his/her own equipment to the communication medium.

Reference Monitors

A reference monitor validates all references to

files, programs, terminals, tapes, or other system resources, which are made by a system user or his program in execution. Validation and subsequent referencing occurs only after the reference monitor checks a data base of access rights and assures that the subject may address the object in the mode (such as read or write) initiated by the subject, as shown in Figure 1.

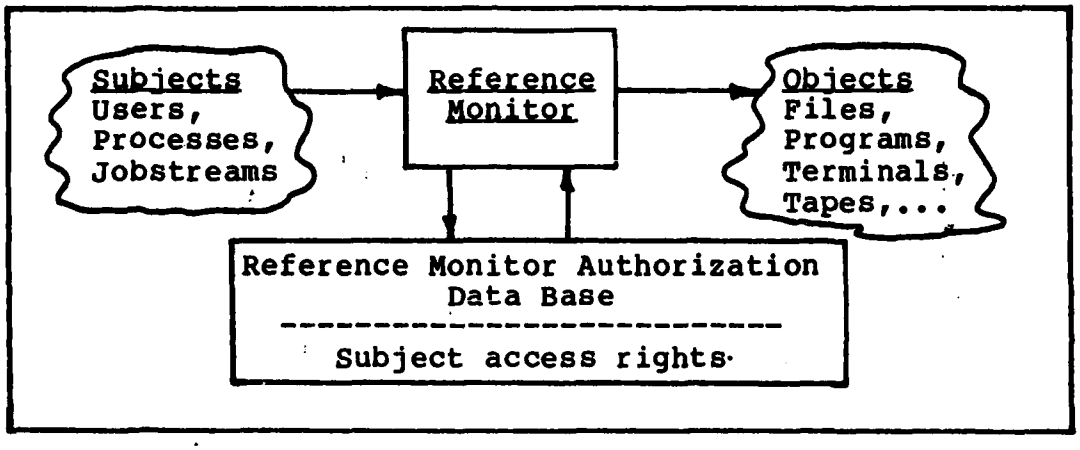


Figure 1. Reference Monitor. (Schacht, 1973)

A computer security technology planning study defined three requirements for the mechanism used to implement the reference monitor. First, the reference validation mechanism must not be subject to unauthorized alteration. This prevents users from aquiring the capability to make unauthorized changes in the relationships of the subjects and objects. Second, the mechanism must be invoked for every reference by any subject to any object. Failure to invoke the mechanism for any reference could breach the system security. Third, the reference validation

mechanism must remain simple enough and small enough that its operation can be completely tested. (Anderson, 1972)

All subject-object relationships could be maintained in an access control matrix. This matrix, as shown in Figure 2, would precisely define the objects each subject may access, and the modes of accessibility. The difficulty imposed with this implementation is the lengthly search time required in a large system due to the sparsity of the matrix.

Subjects	Objects					
	Other Subjects			Files		
	S1	S2	S3	F1	F2	F3
S1		Block Enable		Read Write		
S2			Stop		Update	
S3				Delete	Execute	

Figure 2. An Access Control Matrix. (Anderson, 1972)

Reference monitors can only protect information in the data bases. Other methods must be used to protect information during processing and transferal.

Encryption

Information may be protected when it it transfered or stored by encoding it. The process of encoding the

information is known as encryption. To encrypt information, an encryption algorithm is employed with some key which transforms the information into an unintelligible form. The reverse transformation is called decryption. To decrypt information a key is also required. If the distribution of the keys is limited to those individuals who possess the proper authority to use the information, then the encrypted information is protected from unauthorized access (since no one without a key can understand it).

Conventional encryption algorithms use the same key for encoding and decoding the information. Gerald J. Popek and Charles S. Kline, in their article "Encryption and Secure Computer Networks," describe conventional encryption as a mathematical function,

$$E = F(D,K),$$

where D is the data to be encoded, K is the key to be used, F is the encryption function or algorithm, and E is the resultant encrypted information (Popek, 1979). They explain that an inverse of the function F must exist if the code is to be decrypted since,

$$D = F'(E,K).$$

The algorithms or functions F and F' are only useful if it is extremely difficult to determine D from E without knowing the key used to generate E (and also to regenerate D from E).

Public-key encryption methods use different keys for

encoding and decoding the information. This method derives its name from the fact that the encryption key K can be public knowledge and only the decryption key K' needs to be protected. Public-key methods must insure, however, that it is extremely difficult to determine K' given K, D, and the corresponding E = F(D,K), since this information is public knowledge. The advantage of encryption algorithms using the public-key approach is that anyone can encrypt information using the publicly available encryption algorithm and key, but only those individuals with the private key K' can decrypt the information. This approach may simplify the distribution of matched-key pairs because anyone may encrypt information using the public key for the intended recipient. Only the intended recipient keeps the corresponding private key.

The major difficulty with using encryption to protect information is the handling and storing of keys. Keys must only be given to the proper people and their security must be guaranteed in order to prevent compromise of the information they protect.

Network Protocols

Computers connected in a network can communicate with each other through the use of network protocols. These protocols merely act as a group of translators or interpreters for the computers. The information one computer wants to pass to another computer is translated

into a standard network form by the protocols, transmitted to the desired computer(s), and then translated into a form recognizable to the recipient computer by the protocols. By using a heirarchy of layers, the protocols can be easily implemented and maintained, since each layer only interfaces with adajacent layers. The corresponding layers in separate computers appear to communicate directly with each other and therefore hide the lower layers from the network users. The International Organization for Standardization has developed a Reference Model of Open Systems Interconnections (ISO OSI) (Zimmermann, 1980). This model has seven layers. Andrew S. Tanenbaum describes the function of these layers in his article "Network Protocols" (Tanenbaum, 1981):

(1) The physical layer transmits a raw bit stream into the network and receives the stream from the network.

(2) The data link layer changes an unreliable transmission channel into a reliable one by breaking the raw bit stream into segments, or frames, and checks for errors in the transmission.

(3) The network layer routes packets of frames on the proper channel or line.

(4) The transport layer hides the communications subnetwork from the session layer to provide reliable host-to-host communication.

(5) The session layer initializes, manages, and terminates process-to-process communications.

(6) The presentation layer transforms the communication into a form for the device or file receiving the information.

(7) The application layer's functions are dependent upon the specific uses and applications of the network.

Figure 3 shows these layers for two "host" computer systems in a network and for an intermediate message processor (IMP), which simply forwards or routes messages within the communications subnetwork. Any number of IMPs may be used to complete the connection between the host systems.

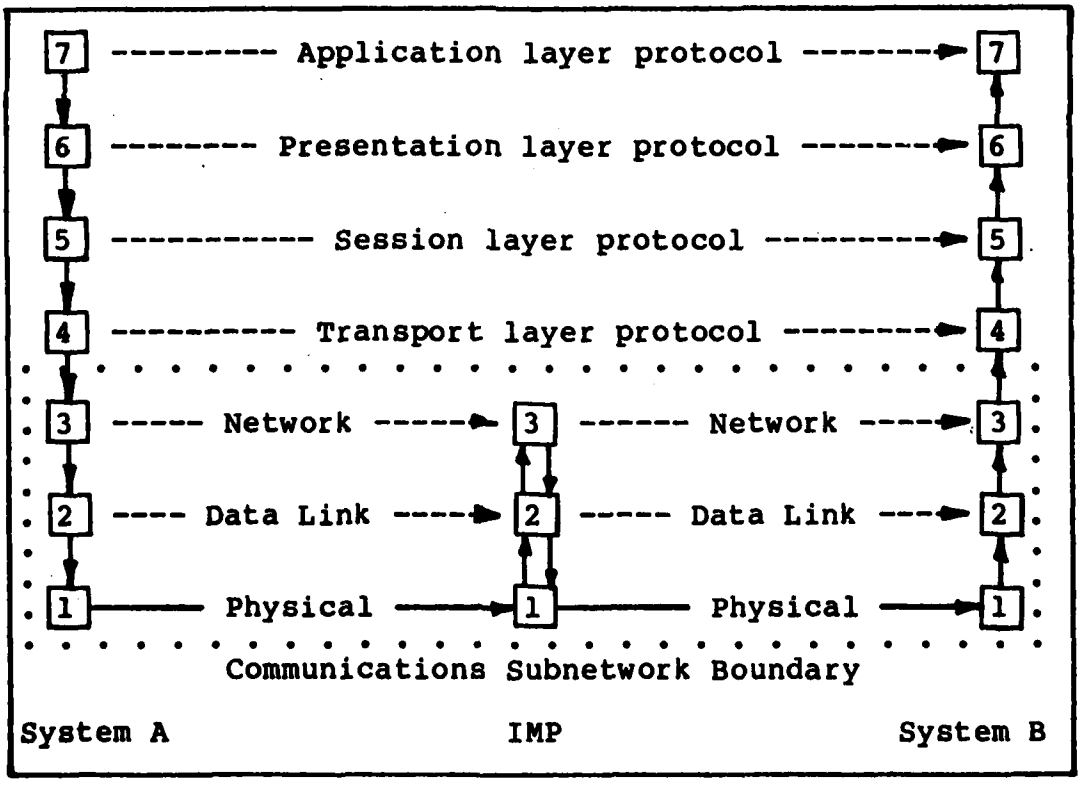


Figure 3. The Seven-layer ISO Reference Model. (Tanenbaum, 1981)

The seven protocol layers are more fully explained the article "Network Protocols," by Andrew S. Tanenbaum (Tanenbaum, 1981). Tanenbaum suggests that encryption should be accomplished in the presentation layer rather than in a lower layer of the protocol as is currently done for link-to-link communications. This would reduce the security requirements for the lower level protocols and would preclude the requirement for the IMPs to be secure since the information passed through the communications subnetwork would be protected by encryption.

Summary

Several concepts must be understood to develop a secure computer network. These include: physical security, necessary to limit physical access to some of the network components; reference monitors, used to validate all references or access attempts to information contained in the network; encryption, used to protect information being transferred over unsecured physical communication media; and network protocols, used to effectively translate and transmit information between computer systems. These concepts can be used to develop a secure computer network.

III Secure Computer Network

This chapter presents the top-level design of a secure computer network. The significance of design development is discussed, followed by the design objective of protecting information. Several implications of this design effort are then presented, indicating the limits and responsibilities inherent in this network model. The information states, or forms in which information may exist in the network, are then discussed. These states are storage, transfer, and processing. A brief explanation of the three major modules which must be developed to secure the information throughout the computer network, or in each of the information states, is then presented. The interrelationships and interactions of these modules is also shown. The top-level design is concluded with the reasons for the development of a logically complete communications subnetwork.

Design Development

The requirement for protecting information in a computer network may be legally, if not morally, imposed. Adequate security must, therefore, be provided for the information contained in or processed by a computer network. Developing a secure environment for electronically processed information is not an easy task. Creating a secure environment for an existing, unsecured computer system is even more difficult, if not impossible.

Security controls for a computer network, therefore, must be designed into the network, rather than simply added to existing networks. The expenditures for the development and operation of such a secure network are only limited by the cost of losing any or all of the information processed by the network. It cannot be overstated that information security is not a nicety in a computer network; it is a necessity.

Design Objective

When developing security for the information in a computer network, the network designer should create an environment which protects every piece of information entering the system from unauthorized use. The term "use" implies a spectrum of capabilities from simply reading the information, to altering or deleting it. If the information entered into the network belongs in the public domain, the network could allow it to be used by anyone with access to the network. Or, more critically, if the information is sensitive or personal, the network should limit the use of information to authorized subjects. The development of a secure network, however, should not diminish the responsibility of the information's custodians to properly identify the subjects authorized to use the information.

Design Implications

The potential network users should realize that once proper information access controls are developed and

implemented, which will indeed secure the information in the network, it will become their responsibility to identify who is allowed to access the information they introduce into the network. Even though a great deal of the access authorization information will be generated automatically, the initial access policies must come from the information's "owners." While the network can limit access to the information in accordance with the wishes or policies of the people who enter that information into the system, the network users should realize that to attempt to secure publically available information in the network would be futile, since anyone could acquire the same information outside of the network.

A secure computer network can only guarantee the security of the information it contains. Once the information is delivered to authorized individuals by the network, it is their responsibility to protect the information. Adequate personnel and procedural security should be implemented to protect the information once it leaves the network.

Information States

To secure information in a computer network merely requires ensuring the security of the information in each of its states within the network. Information within a computer network is always in one of three states: (1) storage, (2) processing, or (3) transfer. Information is considered to be in the storage state after it is

presented to a file or device. Information is in the processing state when it is being manipulated by a processor. Some information in temporary storage may be considered to be in a processing state. Finally, information is in the transfer state while it is being passed from one network component to another. If the information is secure during each of these three states, and the transitions between these states, then the computer network may be considered secure. Figure 4 shows the information states in a simple computer network.

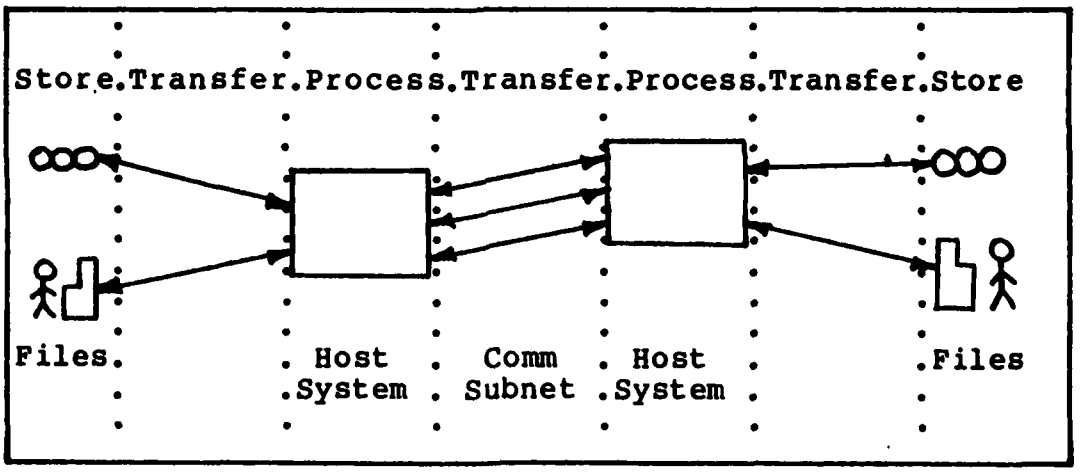


Figure 4. Network Information States.

Providing information security during each of these states is also conceptually simple. With the proper supplement of physical security measures to properly limit the access to key security components, a computer network only requires three essential modules to guarantee the security of the information it maintains or processes:

(1) secure data bases to provide secure long-term storage,

(2) secure operating systems that can maintain the security of the information introduced to the local computer systems from the data bases, peripheral devices, or other network computer systems, and

(3) secure communications between the components of the network to maintain the security of the information being transferred.

Secure Data Bases

Secure data bases ensure the security of the information they maintain through the use of a reference validation mechanism which checks each subject's authorization to access objects within the data base before granting access to those objects requested by the subject. Maintaining strict access control to every object in the data base will protect the information if the subjects requesting the information have been properly identified. Assuming the identification process is correct, the information is secure while it is in the data base, since only authorized subjects are allowed to access it. The problem of proper identification of network users and processes will be discussed later.

Informaticn released from a secure data base must also be protected. Secure operating systems can provide the necessary protection for the information after it is removed from the access control barriers of the secure

data base.

Secure Operating Systems

Secure operating systems assure that the security of information processed by a computer system is not compromised. This is accomplished by designing the operating system so that security controls cannot be circumvented. To do this, the nucleus of the operating system is designed to enforce system security. Minimization of this nucleus is essential to verify its correct implementation. Since all extensions of the operating system are directed by this nucleus, or "security kernel", the system can be verified to be secure (Popek, 1978). Proper physical security may be required to prevent tampering with the security kernel.

The concept of a secure operating system is essential to secure computation in a multilevel or multiuser environment. If the computer is to process information, the information must be in a useable or recognizable form. While the information is in a clear form, outside of the data base access control barriers, it is vulnerable. A secure operating system removes this vulnerability. Without a secure operating system to protect the information while it is being processed, the computer must either be isolated from all external communications and its use restricted to a single user or group of users, or the computer must be manually controlled to insure that no one uses information to which he/she is not entitled.

Either of these choices would exclude the use of the computer in a network.

Information which must be received from or passed to other computer systems in the network, must be requested and delivered through secure communications to prevent unauthorized disclosure.

Secure Communications

Secure communications between computers must be established to prevent the compromise of information transferred between network components. Some transfers can be protected by physically securing the communication medium. For example, a cable between a terminal and a processor, both of which are located in a secure area, would require no further protection if "tapping" the cable was not possible. But, since it is impossible, or impracticle, to physically secure the communication medium between geographically distributed computer systems, secure communications must be established by some other method.

Secure logical channels can be created between subjects using unsecure physical channels by employing encryption (Popek, 1979). If the encryption is employed in the presentation layer of the network protocols, then a secure logical channel can be created between processes or subjects and thereby delete the requirement for physical security of the communications media, including intermediate message processors. The information can then

be transferred securely from one process or subject to another process or subject, or from one secure operating system to another. However, to create the secure logical channel using encryption requires the distribution of a pair of "keys" to the processes or subjects involved in the communication. If the keys were distributed to the actual processes or subjects, the operating systems or individual network components would be required to accomplish many of the network protocol functions, including encryption and decryption. Rather than require so much of the network software to be implemented in each of the network components, endpoints can be provided for the transfer process which translate the transmitted messages into a form suitable for the receipient prior to delivering the message. Not only will these endpoints strengthen the modular concept of the communications network, but they will also allow easier implementation of security and management controls.

Designing a standard of "endpoints" for the communications network will further simplify network development and management. These endpoints would create and destroy secure logical channels from unsecured physical communication channels by using encryption. Thus the transfer of information between these endpoints would be secure, despite the physical security of any communications device. Because these endpoints actually translate the information into a secure form to create the

logical channel, they could easily accomplish the other transformations accomplished by the network protocols. If these endpoints implemented all of the network protocols, they would relieve the computer systems attached to the communications network from accomplishing any "network" functions.

If these endpoints are implemented as independent computers, which are solely responsible for information translation and routing, then the development of a single computer system would produce the major component required to create the secure communications network. These "endpoint" computers, which are simply communications interfaces for geographically distributed, and perhaps dissimilar, computer systems, therefore, contain all of the security required for the communications network. Thus, a physically complete, secure communications network, to which almost any computer system could be connected, may be developed.

Computer systems attached to the communications network could simply hand the endpoints an information request or a piece of information which was bound for another computer system, and the endpoint computers would translate the message into a standard network form and send it to the appropriate system. The endpoint for the receiving system would then translate the communication into a form suitable for the receiving computer and hand it the communication.

If the computer systems connected to the communications network were controlled by secure operating systems, and contained secure data bases, the total information environment would be secure. Each local computer system could view the communications network as a local device or database (a very powerful one), and use it accordingly, without fear of security breeches.

The following chapters more fully develop the functions of the communications network endpoint computers, which are called Secure Network Interface Computers (SNICs).

Summary

Secure computer networks must be designed to protect the information shared by computer systems. This requires the design of a complete environment for the information. All information entering this network environment must be properly marked to identify its authorized users. This marking is the responsibility of the information's custodian even if the function is automated. The information must then be protected throughout its existance in the network.

Information within a computer network is always in one of three major states: storage, transfer, or processing. To protect the information in these three states, and the transitions between them requires three modules in the top-level design: secure data bases, secure

operating systems, and secure communications. All three modules require some physical security measures to ensure their sanctity. To protect the information during transfer between computer systems, we must develop an interface between the communications medium and each computer system in the network, which translates the information into an unintelligble form before transmission, and into a recognizable form after receipt. A computer capable of securely accomplishing these transformations is developed in the following chapters.

IV Secure Network Communications

This chapter presents the conceptual framework for the network communications model presented in the following chapter. The events necessary for a secure exchange of information between two subjects are presented, followed by a discussion of the four phases of the secure communication process: object location, subject identification, request authorization, and request response. The development of network transparancy, the maintanence of a single network directory, and the distributition of access controls to the lowest levels of the network are discussed within the phases of secure communication. These concepts can not only enhance security for the information being handled by the network, but can also simplify network management and control.

Secure Communication Phases

The process of securely transferring information within a computer network is composed of four major phases:

(1) the subject desiring the transfer of information must locate the subject or object which is the source of or destination for the information, and must then contact that subject or object,

(2) the subjects involved in the transfer must identify themselves to each other (or be introduced by a third party) and then verify the identity of the other

subject(s),

(3) one subject must make a request, and the subject(s) of whom the request is made must, before attempting compliance, verify the authorization of the requesting subject to make such a request, and

(4) a response to the request must be made, for which any information which is produced or retrieved is securely transferred and/or stored.

Securely completing this communication process in an unsecured environment can, therefore, be quite a complex task. Not only must the transfer of information be secure, but so too must the identification and request phases. For example, if a subject could add his identification information to the list used by another subject to verify identity, and could also add his name to the list authorizing him to make specific requests, he could request that another subject give him access to information which he is not really authorized to use. It is necessary, then, to protect not only the security of the information, but also the controls which secure that information.

Unless all of the information in the network is to be destroyed to preserve its security, the people and/or machines who are to handle the information must, at some point and to some degree, be trusted. If the method developed to secure information in a computer network can limit the people and components which must be trusted to

individuals and devices which can be trusted, then that method can be considered secure. The primary factor which determines the ultimate security of the information processed by a computer network, therefore, is the trustability of the individuals and devices allowed to handle the information. It is often extremely difficult to determine the reliability of an individual or a device for properly protecting information, because those individuals and devices are so complex. The design of the communications network, therefore, should be as modular, and as simple as possible, so that its correctness, and the correctness of the security method it implements, may be demonstrated. To accomplish this, the phases of secure communication will be considered in greater detail.

Location Phase

Before the resources of a computer network may be used, the subject desiring to use them must locate those resources. Some component(s) of the network must, therefore, be able to locate the desired resources. This location may be accomplished either by broadcasting a request for the resources, or by maintaining a directory of the resources contained in the network. If a broadcast search is used, a message is sent to every network system each time a network component requires the use of resources outside of its local system. This may degrade the network security because the requesting component does not know which system should be responding to the request.

For if the component has no secure way to determine the resource's location, it cannot be sure that the system claiming to possess the resource is indeed the system on which the resource resides. This lack of knowledge allows systems to pose as legitimate sources, whether they are or not. These broadcasts may also swamp the network with unnecessary message traffic. Therefore, a requests for the use of network resources should be made to the subject(s) already known to possess those resources.

The obvious method for determining what subject possesses a particular resource is to maintain a directory of resource locations. This directory can then be used to look up the location of information or devices which network components would like to address. Even though the information in the directory may be public information, the directory is considered the official source of resource location information for the network. This requires that any changes to the directory be carefully controlled to prevent the loss of viable network resources.

To prevent the possibility of removing the required network directory information from the communication network, the directory of resource locations should be maintained by each subject, each system attached to the network, or by the communications network connecting the systems. This will preclude any of the attached systems from dominating the necessary directory information by

providing accessable directory information to each system.

The most appealling approach, initially, is for each subject to maintain its own directory of the resources it is authorized to use. If each subject maintained its own directory, the network would not have any burden for locating information. Rather, each subject would be responsible for knowing the location of every resource it is authorized to use, and for telling the network where that information is located. The difficulty with this aprroach is that it precludes the network from being transparant, and relies on individual subjects to continually update their directories. It is easy to see that automatically updating each subject's directory when an access authorization is changed, or file is deleted could become an extremely complex, if not impossible task. If, however, a directory entry for a particular resource is not updated until a subject using that directory attempts to access that resource, a memory capability of past directory changes is required to ensure that all directories may be properly updated. This essentially requires the maintanence of a central directory, which must be continually referenced. The use of individual directories not only duplicates tremendous amounts of information in the network, but creates the possibility, because of updating problems, that subjects may not be able to find, or even determine, the resources they are authorized to use.

The other two approaches for the location of the network directory are much more managable, and are used in distributed data bases. Either each system maintains a complete directory for the network, or a central network directory is maintained. In either case, the directory should not contain access information, since that is not the directory's function, but should only contain a listing of the network resources and their locations.

If each system maintains a complete copy of the network directory, then some of the same updating problems associated with independent subject directories become evident, and directory currency may become a problem. The number of messages related to directory references, which must be handled by the network would, however, be reduced since each system would have its own copy of the directory.

Maintaining a centralized network directory can enhance security and reliability. The process of updating directory entries is greatly simplified. Since only one copy of the directory is maintained, and because that copy of the directory is directly accessable by all of the network systems, updates may be made quickly and efficiently. The chance of errors in the updating process is also reduced. Only one interaction is required for updating the central directory, as opposed to one interaction with every other system for the locally maintained directories. Using a centralized directory

also ensures that the most current directory information is available, since only one copy of the directory would have to be updated. This may reduce the number of erroneous requests for resources.

For the communications network to be a complete unit, transparant to the systems attached to it, requires that the network directory be maintained within the components of the communications network. This would prevent the requirement for the attached systems to maintain knowledge of the location of other network resources, and would prevent the loss of directory information with the removal of one or more of the attached systems. This requires the directory to be accessed by each communications network endpoint (SNIC) for the subject desiring network resources. Each SNIC may maintain its own copy of the directory, or access a central network directory.

For the remainder of this presentation, a centralized directory will be used. It should be realized, however, that local "caching" or copies of this directory may be maintained by each SNIC, in which case the directory reference scheme would be changed slightly.

Identification Phase

After the subjects involved in a communication are located, they must be introduced and their identities authenticated. Without a proper system of identification authentication, secure information transfer would not be

possible since the recipient's authorization to use the information could not be verified. Therefore, some method of identification must be implemented in any secure computer network. The most preferred method for this identification process would be for the subject possessing the desired resource, and therefore responsible for its security, to directly identify the subject requesting its use. This would preclude the reliance on any intermediary in the identification process. Likewise, the subject requesting the resource must ensure that it is getting the resource requested or at least that it is getting the resource from the subject who owns it and not a fraudulent copy from some imposter. The subject requesting the resource, then, should also directly identify the subject claiming to possess the resource, before accepting the use of that resource.

The placement of responsibility for identification authentication must, therefore, be considered. Identification authentication may be accomplished by either using centralized or decentralized approaches. Centralized approaches, in which all authentication information is maintained in one location, may either use a centralized authority to authenticate identification, or may simply use a centralized library to maintain identification authentication information. Both types require that a centralized data base of all subjects authorized to use any network components be maintained.

In the first case, a centralized authority uses the information in that data base to authenticate an identity and then produces a positive or negative authentication response. In the second case, each system makes its own determination of authenticity using the information in the centralized data base. In either case, individual systems must rely on the centralized data base for authentication, and a security breech in this one location could compromise all of the information in the network.

The second approach to identification authentication is for each system to maintain authenticating information for each individual authorized to use that system. This decentralized approach permits each system to use whatever criteria it deems appropriate for subject identification authentication and removes that systems reliance on a centralized authentication library. While some duplication of information may be present if subjects are authorized to use many systems within the network, the responsibility for the correctness and protection of the authentication information rests with the subject which owns the information. Hence, a breech of security related to identification authentication would only compromise the system which allowed the breech.

Both approaches rely on the correctness of the identification information presented for authentication. It may still be possible to forge "valid" identification credentials which a system may accept and authenticate.

To prevent forgery, or at least make forgery extremely difficult, the information used to establish identity must be unique for each subject, and must be difficult for anyone other than the subject to produce. Many techniques for identifying individuals have been proposed. These include the use of cards, passwords, fingerprints, voice prints, and signature authentication.

Regardless of the identification technique used, the information must be transmitted if it is to be authenticated by a system geographically separate from the location of the subject. To transmit the identification information to the system, it must first be transformed into a digital form. Any information which can be digitally represented may be easily reproduced. To prevent this simple "forgery", attempts are being made to produce a method for establishing identification by using a "digital signature" which is unique for every subject (DeMillo, 1978: 147-168).

Even using digital signature techniques, the system attempting to authenticate a subject must still rely on the agent receiving the identification information from the subject to properly receive and transmit the information. The "agent", or system on which the subject is physically located, must therefore be considered trustworthy by the receiving system if a valid identification is to be established. This "trust" between systems can only be achieved by permitting each system to

be evaluated by the other systems desiring a connection to that system. This evaluation may be accomplished on a periodic basis, by the individuals responsible for the security of the systems. This evaluation is not constrained to be accomplished on-line. If this evaluation is not accomplished for each connection, however, a trust is established between the individuals responsible for maintaining system security. This trust is only established between systems which interact on the network. This implies that there may be some systems connected to the network whose identification information is not trusted by some of the other systems. The information owners must, therefore, be allowed to deny access to subjects requesting information from systems whose identification process is not trusted.

Request Phase

The individual, or group of individuals, who have been trusted to maintain specific resources should authorize any requests to use those resources. Secure operating systems may control the use of system devices or files and secure data bases may control information use by always checking access lists and by carefully controlling who is allowed to change those access lists for specific resources. For example, a company president may not want anyone but himself and the comptroller to be able to authorize access to the company's payroll records. He could assure himself that this is the case if only he and

his comptroller are allowed to make changes to the access authorization controls for that payroll. Additionally, if only one copy of the payroll is maintained on-line, and that is the one controlled by the company president, then updates need only be made to one copy. Anyone desiring information from the payroll would have to retrieve it from the controlled copy. This not only ensures that individuals using the information receive the most current form, but also allows the company president or comptroller to remove or add access authorizations as they see fit. Of course, anyone they allow to use the information may make copies or compromise the information's security. It is important, then, that only trusted individuals are allowed to use the information. Each individual who uses information, therefore, must be responsible for protecting that information while it is in his possession. The "trusted" individuals are also responsible for maintaining the desired level of currency of any files they copy.

Request Response Phase

Once it is determined that a subject is authorized to make a request for the use of a resource, the capability must exist to securely comply with that request. This normally involves the transfer and/or storage of information generated by the request. This information transfer, or storage, must be securely accomplished. If the storage occurs in the same system, the communications

network does not need to be involved.

If information is to be transferred to another system, the information must be secure during movement. In the physical world, the information may be transported by a courier service using an armored car to traverse the distance between the subject and the source. In an electronic environment, however, such physical measures for creating a secure environment for the information during transfer are not possible.

Information may be protected from unauthorized access by using a strong encryption algorithm. If the key, necessary to decode the information, is only known by the intended recipient, then the information may be considered safe since only the intended recipient may transform the information into its original form. The key, like other security controls, must be appropriately protected.

The protection and distribution of these keys, then, becomes the responsibility of the communications network. Several schemes have been suggested for the distribution of keys to the participants in a secure communication. Gerald J. Popek and Charles S. Kline provide an excellent discussion of the basic approaches to key management (Popek, 1978). They also present an approach to establish a secure logical channel using public-key encryption. This method uses a central authority to maintain the current public keys for each system in the network, and to distribute them on request. These keys are then used to

establish a secure logical channel. Incorporation of this approach, slightly modified, with a centralized network directory, and with distributed access controls, yield the model for secure communications presented in Chapter V.

Summary

Transactions, or communications, between the systems of a computer network may be securely accomplished in four phases:

(1) resource location,

(2) subject identification and authentication,

(3) request and request authorization verification, and (4) request compliance or denial.

Resource location can be accomplished using a directory. Maintaining a central directory containing network resource locations and the public keys for each SNIC will strengthen network security, simplify communications management, and enhance the communication network transparancy. This directory may be cached at each SNIC, to reduce message traffic, but cannot be given to the host systems without reducing transparancy and weakening security. Subject identification and authentication must be accomplished to permit secure transfer of information. The communications network should identify and authenticate its endpoints, but it the responsibility of the system resources to authenticate the subjects they respond to. This authentication will permit the approval or denial of the use of a resource by the

individuals responsible for that resource. This distribution of resource access control to the lowest levels of the network, or the actual owners of the network's individual resources (files and devices), enhances the security and control of those resources, because it places the responsibility for information security squarely on those individuals trusted to maintain the information. Finally, the security of any information transferred during the transaction must be securely accomplished by the communications network. The precise method for accomplishing this transfer will be developed in Chapter V. Upon completion of the transfer, the information's security becomes the responsibility of the trusted subject(s) receiving it.

V Secure Communication Model

This chapter presents a secure network communication model. The model components are identified and then the method used to establish and conduct secure communications is presented. From this method, the functions required of each component are identified. These functions should completely define the requirements for each component and should therefore fully specify the model.

Components

Four major types of components comprise this secure network model. They are:

(1) Secure Data Bases (SDB), which provide secure storage and retrieval of information,

(2) Kernelized Secure Operating Systems (KSOS), which provide a secure processing environment for the information after its removal from the SDB or receipt from peripheral device or other system,

(3) Secure Network Interface Computers (SNIC), which serve as endpoints for the communications network and perform all of the network communication functions as a "front-end" processor for the attached host system, and the

(4) Network Directory and Security Center (NDSC), which maintains the location of all network resources and the public keys for those resources' respective SNICs.

These components are shown in Figure 5.

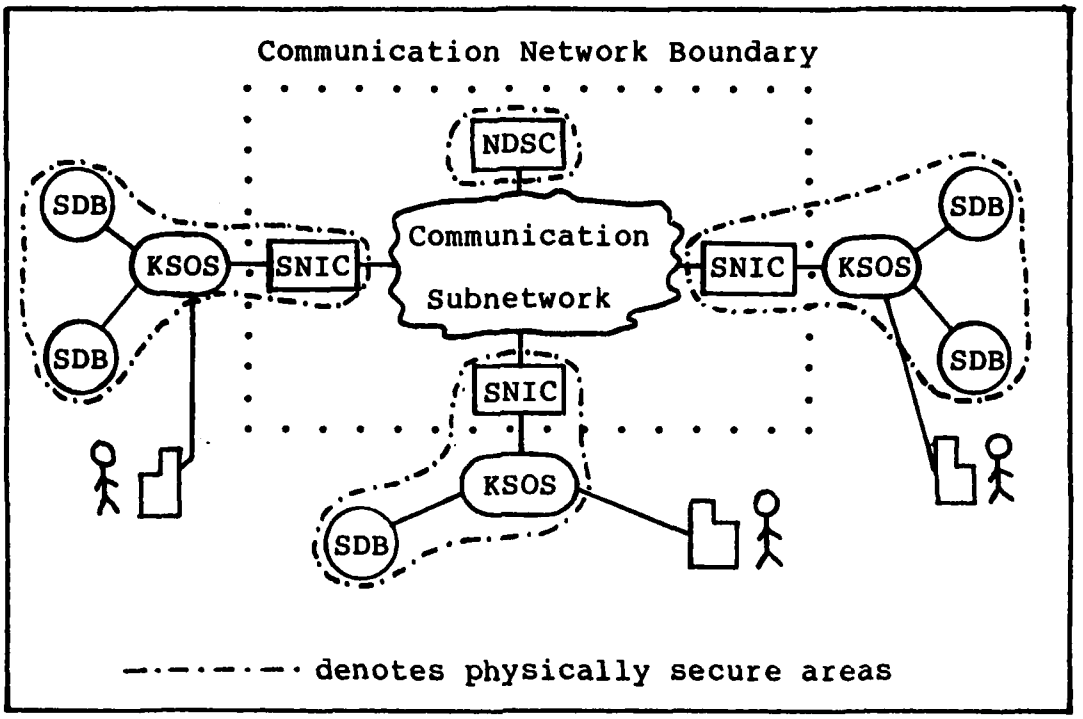


Figure 5. Secure Network Model.

The SNIC and the associated KSOS must be in a physically secure environment to ensure the sanctity of their security controls. This allows them to be connected via a physically secure cable and eliminates the possibility of "tapping" the communication media between them. The NDSC must also be in a secure environment to protect the directory information from unauthorized tampering. The exposure of resource locations or public keys is not a security problem, since this information may be public knowledge. The physical protection does, however, help prevent unauthorized changes to the information which may deny the use of network resources by hiding their location from other network systems.

Limiting the use of the directory to authorized systems (systems with public keys) may, however, provide an additional obstacle for illegitimate users attempting to probe the system for information.

The components of this secure network model may interact to provide a secure environment for the information introduced into this network. For secure communication to occur, each SNIC must maintain two keys: the public key of the NDSC (P_{NDSC}), and the private, or secret key, (S_{SNIC}) corresponding to its own public key. For example, $SNIC_A$ needs to maintain P_{NDSC}, the public key of the NDSC, and S_A, the secret key of $SNIC_A$. The NDSC must maintain the public key of each SNIC, P_A, P_B, ..., and its own secret key, S_{NDSC}. With only these keys initially known to the SNICs and the NDSC, the following secure communication method may be used.

Secure Communication Method

Suppose a subject, S1, who is working on System A, wants to access a network file, NF1, not located on the same system. The following sequence, as shown in Figure 7 on page 71, will occur:

(1) S1 makes a request to $KSOS_A$ to use file NF1.

(2) $KSOS_A$ discovers the file is not in the local system and passes S1's request to $SNIC_A$ with the identifiers for the subject, S1, and the device from which the request was made, D1.

(3) $SNIC_A$ queues the request until a secure logical

channel can be established with NF1. To establish this channel, $SNIC_A$ must first locate NF1. $SNIC_A$ therefore asks the NDSC for the location of NF1. This request for the location of a network resource includes the type of message, RL, the requesting SNIC's identifier, A, the identification of the resource requested, NF1, and a unique identifier, IDA1, which will be used to determine the "currency" of the response and will prevent a "playback" of an old response by an intruder between the SNIC and the NDSC. This entire message is encrypted in the public key of the NDSC. The message would therefore be in the form:

$$[RL, A, NF1, IDA1]^{P}NDSC.$$

(4) The NDSC decrypts the message using S_{NDSC}. It then looks up the location of NF1 and discovers that NF1 is located on System B. The NDSC then looks up the public key for System B and responds to the request for the location of NF1 made by $SNIC_A$. The response includes the type of message, LN, the identification of the system generating the response, NDSC, the resource location, B, the public key for that location, P_B, and the unique identifier included in the request message, IDA1. This identifier satisfies $SNIC_A$ that the response is current and that the response came from the NDSC, since only the NDSC has S_{NDSC} and is therefore the only system that could have decrypted the request and unique identifier. The entire response is encrypted in the public key of $SNIC_A$

and is in the form:

$$[LN, NDSC, B, P_B, IDA1]^{P}A.$$

(5) $SNIC_A$ receives the message and decrypts it with its own secret key. It then knows the location of NF1 and the public key for the SNIC of the system containing NF1. Because the unique identifier, sent to the NDSC with the request for the location of NF1, has been correctly decrypted by the NDSC, $SNIC_A$ is satisfied that the information is actually from the NDSC. $SNIC_A$ must now contact $SNIC_B$ to begin establishing the logical channel between the subject, S1, and the object, NF1. $SNIC_A$, therefore, sends a message to $SNIC_B$ with the following information: the type of the message, RN, the identification of the message originator, A, the desired resource, NF1, the identifier of the subject desiring to use that resource, S1, the device on which that subject resides, D1, and a unique identifier, IDA2, to establish the currency of $SNIC_B$. The entire message is encrypted with the public key of $SNIC_B$. The message is in the following form:

$$[RN, A, NF1, S1, D1, IDA2]^{P}B.$$

(6) $SNIC_B$ must decrypt the message using its secret key. After decryption, $SNIC_B$ requests the public-key of the message's originator, $SNIC_A$, from the NDSC. The public key is obtained from the NDSC rather than included in the message from the requestor to prevent an unauthorized system from acting as an authorized one and

providing its own public key for the channel. The NDSC, therefore, serves as an authentication mechanism for valid systems since it only maintains authorized public keys. The key request message includes the message type, RK, the identifier of the message originator, B, the identifier for whom the public key is desired, A, and a unique identifier, IDB1. The message is encrypted in the public key of the NDSC and is in the form:

$$[RK, B, A, IDB1]^{P}NDSC.$$

(7) The NDSC receives the request from $SNIC_B$, decodes it using its own secret key, and looks up the public key of $SNIC_A$. It responds to the request for the key with a message including the message type, KN, the identification of the system generating the response, NDSC, the identifier of the system of the key request, A, the public key requested, P_A, and the unique identifier sent to it by $SNIC_B$ in the key request message, IDB1. The entire response is encrypted with the public key of $SNIC_B$. The response is in the form:

$$[KN, NDSC, A, P_A, IDB1]^{P}B.$$

(8) $SNIC_B$ receives the response from the NDSC and decrypts it using its own secret key. $SNIC_B$ is satisfied of the authenticity and currency of the NDSC since the message it sent was decrypted with the secret key of the NDSC, which is only known to the NDSC, and because the response contained the unique identifier included in the request message. $SNIC_B$ must now authenticate itself to

$SNIC_A$ and request it to authenticate. It may also begin establishing the secure logical channel for the process-to-process communication between S1 and NF1. The message sent to $SNIC_A$, therefore, includes the message type, AN, the identifier of the source of the message, B, the unique identifier, IDA2, sent to $SNIC_B$ by $SNIC_A$, another unique identifier, IDB2, which $SNIC_B$ will use to authenticate $SNIC_A$, and the key, K1, which will be used to encrypt all communications between S1 and NF1. The entire message is encrypted in the public key of $SNIC_A$, so that only $SNIC_A$ can interpret it. The message is in the form:

$$[AN, B, IDA2, IDB2, K1]^{P}A.$$

(9) $SNIC_A$ receives and decodes the message using its secret key. $SNIC_A$ is now sure that $SNIC_B$ is authentic and current since it decoded the message sent by $SNIC_A$ and its response included the unique identifier sent in that message. $SNIC_A$ also has the key which will be used in the communication between NF1 and S1. $SNIC_A$ may now send S1's original request, which has been queued at $SNIC_A$, to NF1 through $SNIC_B$. $SNIC_A$ therefore encrypts the request with K1, the key for the secure logical channel between S1 and NF1. Since $SNIC_A$ has not yet authenticated itself to $SNIC_B$, it must also include that authentication in this message. $SNIC_A$ therefore sends $SNIC_B$ a message which includes the message type, AR, the identification of the message originator, A, and the unique identifier, IDB2, sent to it by $SNIC_B$. The message is encrypted with the

public key of $SNIC_B$ and is in the form:

$$[AR, A, [request]^{K1}, IDB2]^{P_B}.$$

(10-11) $SNIC_B$ receives the message and decodes it using its secret key. $SNIC_B$ is now satisfied that $SNIC_A$ is authentic and current, since it decoded the message with its secret key and returned the unique identifier. The secure logical channel has been established and S1's request has been received. This request is decrypted using the secure logical channel key, K1, and is forwarded to NF1 through $KSOS_B$, along with the subject identifier and the device identifier.

(12-13) NF1 is under no obligation to fulfill the request. In fact, NF1 does not even have to accept the subject and device identification provided to it by the network. NF1's response may be a request for further identification from S1, or may simply fulfill or deny the request. NF1's response to the request is returned to $SNIC_B$ through $KSOS_B$.

(14) $SNIC_B$ encrypts the response in K1 and completes the message by adding the message type, NI, its own identifier, B, and the unique identifier for the secure logical channel selected by $SNIC_A$, IDA2. The entire message is encrypted in the public key of $SNIC_A$ and is in the form:

$$[NI, B, [response]^{K1}, IDA2]^{P_A}.$$

(15-16) $SNIC_A$ receives the message and decrypts it using its own secret key. It uses the unique secure

channel identifier to retrieve the channel key K1. $SNIC_A$ then decrypts the response, adds the necessary identifying information, and passes it to S1 through $KSOS_A$.

All subsequent communication between S1 and NF1 may be conducted with the established secure logical channel. The subsequent communications between the subject's SNIC and the object's SNIC are sent in the form of message 14, which is an NI type message.

If the total message length of the channel (the sum of the lengths of all of the messages encrypted in the same key) exceeds a specified limit, the key must be changed. This prevents extreme message lengths from revealing statistical properties of the language and compromising the validity of the encryption. If the maximum allowable channel length has been exceeded, a message is sent to the other SNIC controlling that channel to change the key for that channel. This message includes the message type, KC, the unique identifier for the channel used by the receiving SNIC, like IDB2, and the new key, K2. The message is encrypted in the public key of the receiving SNIC and is in the form:

$$[KC,\ IDB2, K2]^{P}A$$

If a SNIC's public key needs to be changed, that SNIC must notify the NDSC and the other SNICs to which it has currently established channels. The NDSC must be notified so that the directory may be changed. The directory update method, described in the next section, may be

employed to make this change. The other SNIC's may be notified through one of the existing channels. The public-key change message, type KS, includes the new key, P_A, and the unique identifier of an existing channel with the receiving SNIC, like IDB2. The entire message is encrypted in each receiving SNIC's public key, and is in the form:

$$[KS, P_A, IDB2]^{P_B}$$

Each SNIC receiving this message would then use the new public key for all channels connected to the SNIC which transmitted the message. Because the message is sent by the SNIC over an existing channel, the possibility of fraudulent key change messages is eliminated (unless a channel has already been sucessfully infiltrated).

The secure logical channel will be terminated by the SNICs when one of three conditions occur: timeout, notification from the other SNIC, or notification from the KSOS. If the KSOS notifies the SNIC that a process has terminated, all channels established for that process may be terminated. In addition, if a channel remains inactive for some specified period of time, that channel should be terminated to prevent the maintanence of unused channels. In either case, the other SNIC(s) controlling the channel(s) should be notified. To do this, a termination message is sent. It includes the message type, TN, the identifier of the SNIC generating the message, like A, and the unique identifier used for that channel by the other

SNIC, like ··R2. The message is encrypted in the public key of the receiving SNIC and is in the form:

$$[TN, A, IDB2]^{P}_{B}$$

Upon receipt of this message the channel identified would be terminated by removing it from the CST.

Directory Update Method

Network Directory updates are required when a network resource is added to or removed from the network, or when the public key of one of the SNICs is changed. The process for updating the central directory is very simple. When a network resource is added or deleted, the responsible KSOS tells its SNIC. The SNIC sends a message to the NDSC to either add or delete an entry for that resource. The SNIC generates its own message when its public key is changed. The directory update message must include the SNIC's identification, the type of the message, the type of update, the resource to be updated, and a unique identifier for a currency check. The entire message is encrypted in the public key of the NDSC. The message is in the form:

$$[DU, A, delete, NF1, IDA3]^{P}_{NDSC}.$$

Before the directory change is made, the NDSC must authenticate the SNIC. The NDSC returns a message which includes the message type, the identification of the NDSC, the unique identifier sent by the SNIC, and a second unique identifier used to ensure the SNIC is current. The message is encrypted in the public key of the SNIC and is

in the form:

$$[AU, NDSC, IDA3, IDN1]^{P_A}$$

The SNIC decodes the message and using its secret key and checks the unique identifier to verify the authenticity and currency of the NDSC, since only the NDSC could have decrypted the DU message containing the unique identifier. The SNIC then sends the authentication response, encypted in the public key of the NDSC, in the form:

$$[AD, A, IDN1]^{P_{NDSC}}.$$

The NDSC then updates the directory item. This directory update sequence is shown in Figure 6.

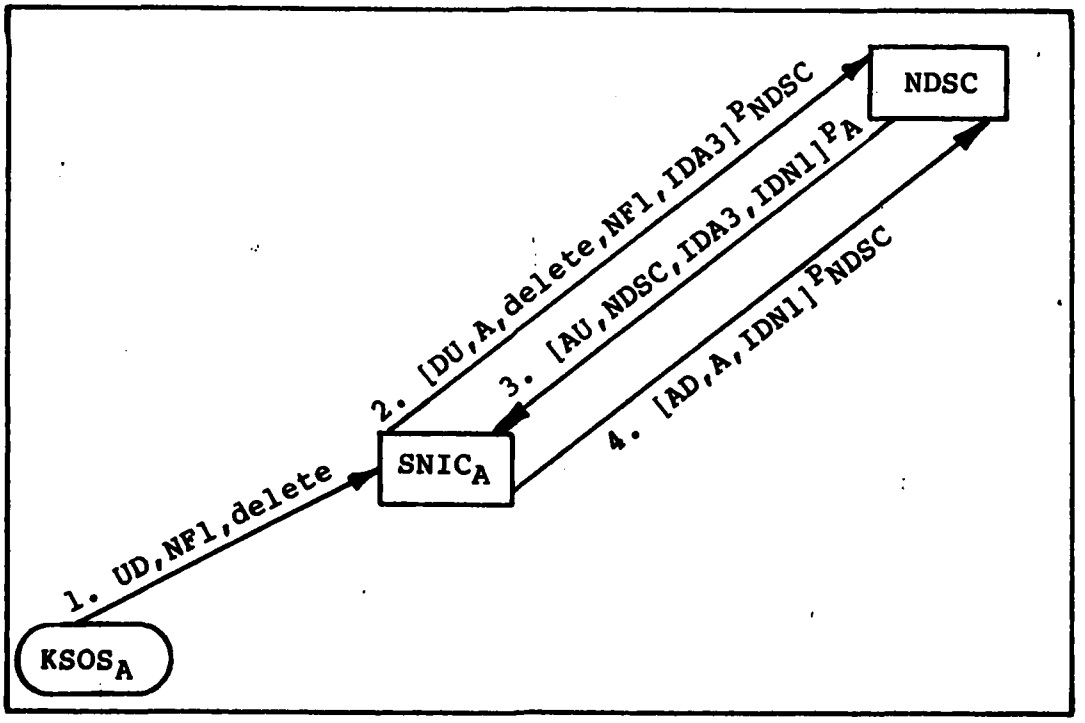


Figure 6. Directory Update Method.

SNIC Functions

The Secure Network Interface Computer is a front-end processor responsible for all of the network protocols. Layer 6, the presentation layer, must transform the information presented to it by the KSOS into the standard network form. This includes marking the message with one of the standard network message types which may be received from the KSOS. The three type of messages the SNIC (layer 6) may receive from the KSOS are: directory updates, process termination notifications, and messages containing information which is bound for other network resources. These message types are identified in Table I on page 60. After translating the message into standard network form, layer 6 gives the message to layer 5. Layer 6 must also transform information coming from the communication network (layer 5) into a form suitable for the KSOS. The implementation of layer 6 will therefore depend on the specific system attached to the SNIC. A different version of layer 6 is required for each different type of system used in the network.

Layer 5, the session layer, performs the necessary management functions and implements security for the process-to-process communications. This requires that layer 5 be able to perform the following functions:

(1) Build messages to initiate, conduct, and terminate communications for the involved processes.

(2) Generate unique identifiers for the secure

logical channels.

(3) Generate keys for the secure logical channels.

(4) Encrypt information with the keys.

(5) Decrypt messages with the appropriate keys.

(6) Monitor the total message length encrypted with each key.

(7) Monitor the time a secure channel remains in use.

(8) Add "headers" to messages for the upper and lower levels to indicate the destination of the message.

(9) Generate new sets of public/secret keys to replace its own public key when required.

(10) Maintain information to manage each secure channel.

(11) Transfer messages in both directions between layer 6, the presentation layer, and layer 4, the transport layer.

The specific functions performed for each message received by layer 5 are dependent on the direction of the flow of the information and the type of message. For messages from the communications subnetwork bound for the KSOS (going from layer 4 to layer 6), the following procedure is accomplished:

(1) Decrypt the message using the current secret key of the SNIC.

(2) Check the message type and perform the appropriate functions for that message type, as specified

later in this section.

For messages from the KSOS which are bound for the communications subnetwork (going from layer 6 to layer 4), only the second step is required, since the information is not encrypted.

Layer 5 must maintain information about the process-to-process communications in progress. This information may be maintained in a Communications State Table (CST). This table should contain the following information about each process-to-process commmunication in progress:

(1) the identifier of the subject requesting the resource,

(2) the identifier of the device the subject is using,

(3) the identifier of the network resource the subject wants to use,

(4) the location of the resource or the subject requesting the resource (depending on which SNIC's CST is being discussed),

(5) the public key of that location's SNIC,

(6) a unique identifier for the secure logical channel, to be used by the subject's SNIC,

(7) a unique identifier for the secure logical channel, to be used by the resource's SNIC,

(8) the key for the secure logical channel, and

(9) the last time the channel was active.

This information is not all available at the

beginning of a communication, but is only entered as required. The public key of the resource's location may be kept in a different location, to prevent duplicating information within the table. The other information must be maintained for each process-to-process communication in progress.

Table I. Communications Network Message Types.

Message Type		From	To
RL	Request for Resource Location	SNIC	NDSC
LN	Location of Network Resource	NDSC	SNIC
RN	Request to Create Logical Channel	SNIC	SNIC
RK	Request for a Public Key	SNIC	NDSC
KN	Public Key for a SNIC	NDSC	SNIC
AN	Authentication Reply and Key	SNIC	SNIC
AR	Authentication Reply and Request	SNIC	SNIC
DU	Directory Update	SNIC	NDSC
AU	Authentication for Update	NDSC	SNIC
AD	Authentication Reply to Directory	SNIC	NDSC
NI	Network Information	SNIC	SNIC
TN	Termination of a Logical Channel	SNIC	SNIC
TP	Termination of a Process	KSOS	SNIC
UD	Directory Update	KSOS	SNIC
PI	Process Information	KSOS	SNIC
KC	Key Change for a Logical Channel	SNIC	SNIC
KS	Key Change for a SNIC's Public Key	SNIC	SNIC

Table I lists the message types which may be present in the communication network. These include messages passed between SNICs, between the SNICs and the NDSC, and between the KSOSs and the SNICs. References to the SNIC really implies layer 5 of the SNIC protocol. Reference to the KSOS really implies layer 6 of the SNIC protocol.

For each message type, certain functions are required of the component receiving that message. These functions, performed in layer 5 of the protocol, are described below. If, at any time during the processing of these messages, any irregularities are discovered, an auditing procedure is called which records the problem encountered and the message involved. Further processing of such messages is terminated.

PI type messages require the SNIC to:

(1) Check the CST to locate an entry for the subject, device, and resource of the message. If an entry is not found, no secure channel exists for this process-to-process communication, therefore:

a. Make an entry in the CST for this process, initially filling only the subject, device, resource, location, public key, unique channel identifier, and time. Use the NDSC as the location and also use its public key.

b. Build an RL message using this table entry.

c. Encrypt the RL message in the public key of the NDSC.

d. Add the necessary header information.

e. Give the message to layer 4.

(2) If an entry is located in the table, a secure logical channel already exists for this process-to-process communication. In this case, check the CST to determine if the channel is completely established. If the channel is still being established, queue the message to allow the channel to be completed for this process-to-process communication. If the channel is completed, then:

a. Update the CST channel entries for time and total message length.

b. If the total message length exceeds the specified limit, then:

1. Put a new channel key in the CST.

2. Build a KC message from the CST.

3. Encrypt the message in the public key.

4. Add the necessary header.

5. Pass the message to layer 4.

c. Encrypt the message with the channel key.

d. Build an NI message.

e. Encrypt the message with the public key of the destination SNIC.

f. Put a header on the message.

g. Give the message to layer 4.

UD type messages require the SNIC to:

(1) Make an entry in the CST which includes the

subject, device, resource, resource location (NDSC), public key of the NDSC, unique identifier for the update, and the time.

(2) Build a DU message.

(3) Encrypt the DU message in the NDSC's public key.

(4) Add a header to the message.

(5) Give the message to layer 4.

TP type messages require the SNIC to:

(1) Look up the entry for the referenced subject, device, and resource.

(2) Build a TN message.

(3) Encrypt the TN message in the public-key of the other SNIC.

(4) Add a header to the message.

(5) Delete the entry from the CST.

(6) Give the message to layer 4.

LN type messages require the SNIC to:

(1) Find the entry in the CST by using the unique identifier.

(2) Enter the location of the resource, the public key of its SNIC, and the time in the CST.

(3) Generate a new unique identifier and enter it into the CST.

(4) Build an RN message using the CST entry.

(5) Encrypt the message in the public key of the resource's SNIC.

(6) Add a header to the message.

(7) Give the message to layer 4.

RN type messages require the SNIC to:

(1) Make an entry in the CST for this new process with the subject, device, and resource identifiers, the identifier of the other SNIC, its unique channel identifier, and the time.

(2) Generate a unique identifier for this channel and enter it into the CST.

(3) Build an RK message.

(4) Encrypt the message in the public key of the NDSC.

(5) Add a header to the message.

(6) Give the message to layer 4.

KN type messages require the SNIC to:

(1) Find the entry in the CST using the unique identifier.

(2) Enter the public key and time in the CST entry.

(3) Generate a new unique identifier and channel key and enter them in the CST for this logical channel.

(4) Build an AN message.

(5) Encrypt the message using the subject SNIC's public key.

(6) Add a header to the message.

(7) Give the message to layer 4.

AN type messages require the SNIC to:

(1) Find the entry in the CST using its own unique identifier for this secure logical channel.

(2) Enter the resource SNIC's unique identifier, the channel key, and the time in the CST entry for this channel.

(3) Build an AR message.

(4) Encrypt the message in the public key.

(5) Add a header to the message.

(6) Give the message to layer 4.

AR type messages require the SNIC to:

(1) Find the entry in the CST using the unique identifier and update the time.

(2) Decrypt the request using the channel key.

(3) Add the necessary identifying information to the request, to include the subject, the device, and the destination resource.

(4) Give the message to layer 6.

AU type messages require the SNIC to:

(1) Find the entry in the CST using the unique identifier in the message and update the time.

(2) Build an AR message.

(3) Encrypt the message in the public key of the NDSC.

(4) Add a header to the message.

(5) Give the message to layer 4.

NI type messages require the SNIC to:

(1). Find the entry in the CST using the unique identifier.

(2) Update the time and total message length for the

channel entry in the CST.

(3) Decrypt the message using the channel key.

(4) Add identifying information to the message, including the subject, device, and resource identifiers.

(5) Give the message to layer 6.

KC type messages require the SNIC to:

(1) Look up the entry in the CST using the unique identifier.

(2) Enter the new channel key and the time in the CST entry for that channel.

KS type messages require the SNIC to:

(1) Look up the entry in the CST using the unique identifier.

(2) Determine the other SNIC controlling that channel.

(3) Change the public key of every channel connected to that SNIC.

TN type messages require the SNIC to:

(1) Look up the entry in the CST using the unique identifier.

(2) Delete the entry.

After layer 5 processes messages bound for other systems, it gives them to layer 4. Layer 4 is responsible for transporting the message to the proper location. It hides the actual communications medium from the higher layers. It breaks the message into packets, in packet-

switched networks, and reassembles them at the receiving location. It provides a reliable communication system for the network. Layers 1 through 4, may be implemented in any manner desired, since the information contained in the messages they transport may only be transformed into a recognizable form by the intended recipient. This permits any type of communication medium to be used. Problems related to message concentrators, routing, message assembly, message acknowledgement, and actual transmission are issues to be solved in developing the layers 1 through 4. These problems do not, however, affect the security of the information contained in the messages. Security of the information and reliability of the transport mechanism are now two separate issues. If the transmission system is not reliable, however, use of the information may be denied to authorized subjects, but its security will not be compromised.

NDSC Functions

The Network Directory and Security Center (NDSC) maintains a listing of all network resources and their locations. It also maintains a list of the public keys of all of the SNICs in the network. The NDSC contains the same first four layers of the network protocols as the SNICs. The fifth layer's basic functions are the same, but the message types which must be processed are different from the types processed by the SNICs. Layer 5 of the NDSC's protocols performs the following steps:

(1) Decode incoming messages using S_{NDSC}.

(2) Determine the message type.

(3) Perform the appropriate functions for that type.

RL type messages require the NDSC to:

(1) Look up the resource location in the directory.

(2) Build an LN message.

(3) Encrypt the message in the public key of the requestor.

(4) Add a header to the message.

(5) Give the message to layer 4.

RK type messages require the NDSC to:

(1) Look up the key in the directory.

(2) Build a KN message.

(3) Encrypt the message in the public key of the requestor.

(4) Add header information to the message.

(5) Give the message to layer 4.

DU type messages require the NDSC to:

(1) Make an entry in the Directory Update Table (DUT), to include the source of the update, the type of the update, the resource affected or the new key if it is a key change, the unique identifier of the source, and the time.

(2) Look up the location of the resource in the directory.

(3) Compare the source of the update with the location of the resource to ensure the source is

authorized to make the update.

(4) Generate a unique identifier for the update and enter it into the table.

(5) Build an AU message.

(6) Encrypt the message in the public key of the source.

(7) Add header information to the message.

(8) Send the message to the source of the update.

AD type messages require the NDSC to:

(1) Look up the entry in the DUT using the unique identifier.

(2) Make the requested change in the directory.

(3) Delete the entry in the DUT.

Summary

Secure communication between network components is possible through the use of the model presented. The model is comprised of four major type of components: SDBs, KSOSs, SNICs, and an NDSC. These components may interact to provide security for the information transmitted through the network. The method used in this model establishes a secure logical channel for each process-to-process communication. These channels are managed by layer 5 of the network protocol. Layers 1 through 4 are responsible for the reliability of information transfer, but not the security. Layer 5 maintains information security for each channel by encrypting the information it receives for that channel in

a key used only in that channel. Layer 6 provides the interface with the host computer system and transforms information to and from the standard network form for that host system. The host KSOS must maintain the separation of information for the processes it supports. The NDSC provides an easily accessable, readily available source of network resource location information as well as the public keys for each of the network SNICs. The SNICs provide a standard, secure communications network for the KSOS computer systems attached to them.

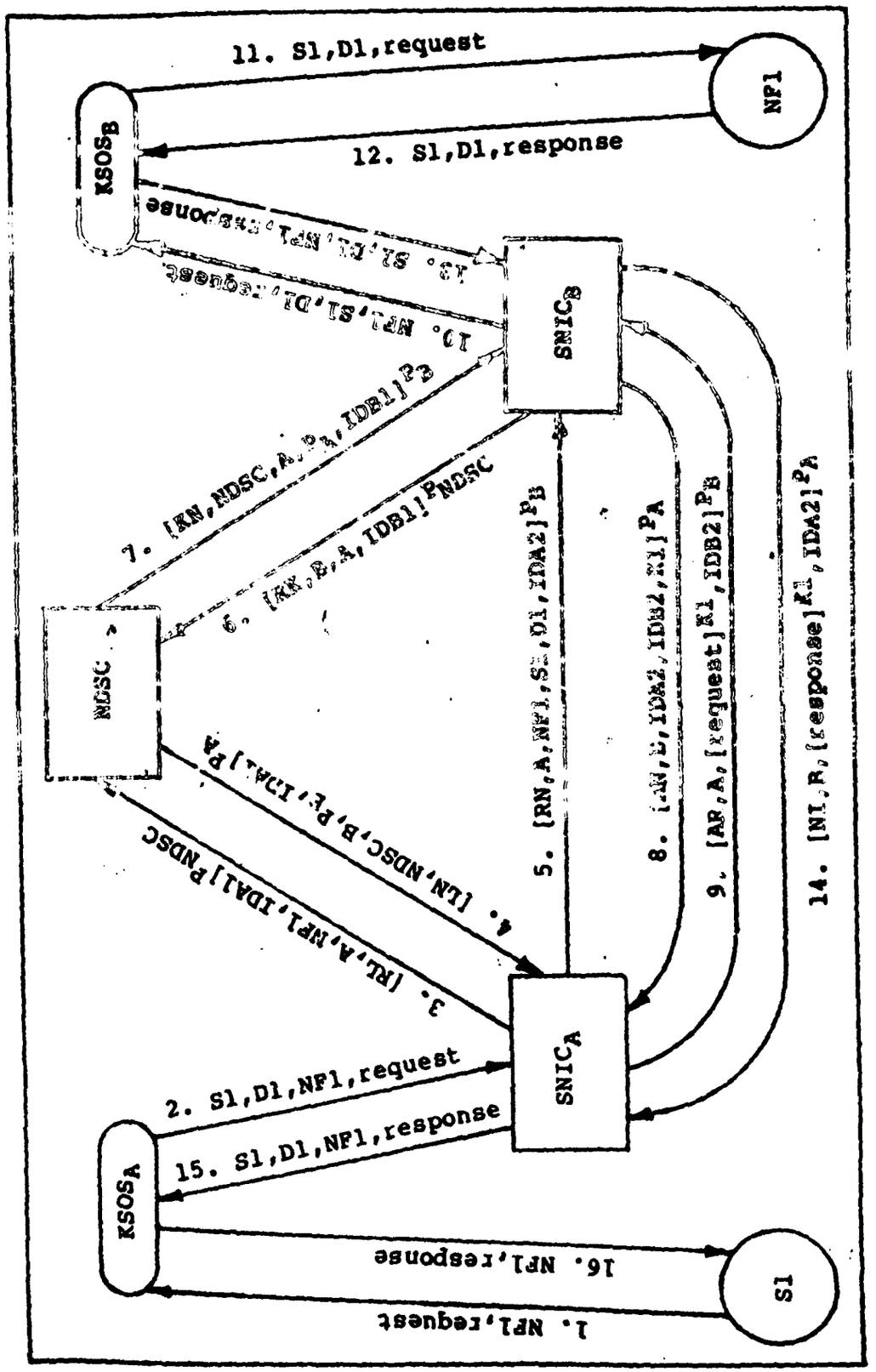


Figure 7. [illegible] Communication Method.

VI Analysis of Secure Communications Model

This chapter presents an analysis of the secure communications model developed in the preceding chapter. First, a finite state analysis of the communications channels is completed. This analysis not only demonstrates the integrity of the model, but also identifies the factors which control the probability of compromising information security. Then, a discussion is presented which identifies areas of consideration for the detailed design, implementation, certification, operation, and management of such a secure network. These areas of consideration are not intended to encompass every detail or even every major topic which must be considered for a full scale development effort, but rather are included to provide insights (derived during the development of a simulation of the secure network model) which may be beneficial during a full scale development of the network.

The secure communications model specified in the preceding chapter has been simulated using a SLAM II Combined Simulation. The simulation modelled a communications network which consisted of three Secure Network Interface Computers (SNICs) and a Network Directory and Security Center (NDSC). A kernelized secure operating system (KSOS) was attached to each of the SNICs. This simulation was developed to demonstrate the feasibility of the design, to stimulate thought on

implementation difficulties of the design, and to provide a prototype tool for further study of the network's operation. The simulation was not used to provide information on the network overhead for security or security costs (even though it could be used to do so if realistic time delay distributions for message traffic in the network are known), but rather primarily served to substanstiate the validity of the model and permit preliminary analysis of implementation problems.

Finite State Analysis of Communication Channels

The secure communications network model developed in this thesis can be shown to be secure. Furthermore, it can be shown that the probability of bypassing the security barriers is dependant upon the length of the keys and identifiers used to specify the secure logical channels. Since the length of these keys and identifiers can be controlled, so too may the probability of information compromise. The assumptions upon which this demonstration of security is based are:

(1) Secure operating systems exist and are used to implement the SNICs and the NDSC.

(2) The SNICs and the NDSC are physically secure.

(3) The communication medium connecting each host system's operating system and that host's SNIC is secure.

(4) The secret key of the NDSC and each SNIC is secure.

The communications network can be shown to be secure

when channels can be established between two authorized endpoints which are capable of passing information which is unintelligible to anyone other than authorized recipients (normally those two endpoints). These secure logical channels are not dependent on the security of the actual physical medium used to transport the information. To insure that a secure logical channel is actually established between two authentic endpoints, the channel must be developed in a manner that can be verified to be secure, or that at least can identify the probability of compromise for those steps in the process which may not be absolutely safe. This can be done if the channel is considered to progress through a specified series of states until a secure logical channel is developed. If this progression through the states can be shown to be secure, and the probability of illegal transitions can be identified, then the channel may be considered secure (within the specified probability). In addition, it must be demonstrated that the process of establishing a secure channel will not permit states to be bypassed to reach a state which is considered secure.

Because the channel states are maintained in a file (CST) which is only accessable through the secure operating system of the SNIC, the states may only be changed by the processes or subjects which are authorized to use that file. If only the message processing program in layer 5 of the SNIC is allowed to use that file, the

channel states are protected from unauthorized changes. Therefore, one must only demonstrate that the authorized state changes made by layer 5 of the SNIC are made in a logically sound manner that will not permit unauthorized parties to become involved.

The model presented in the preceding chapter only permits a channel to transition to the next authorized state if a message is received which contains the proper control information. Additionally, only certain types of messages may be processed in each state. If any of the control information (message type, keys, and identifiers) contained in a message is found to be invalid for the current state of the channel, an auditing procedure is called which identifies the message and the discrepancy to system managers. The control information within the messages is protected because it is encrypted in the public key of the authorized recipient when it is transmitted. Therefore, the secret key of the recipient is required to obtain the information, and since it is known only by the authorized recipient, the control information is protected during transmission between SNICs. The probability of the SNIC allowing fraudulent messages to pass undetected through the network is the probability of guessing the required control information. This control information may include the public key of the receiving SNIC, the message type, the unique identifier the receiving SNIC has selected for the channel, and/or

the key used to encrypt the actual information passed in the channel.

Each SNIC can identify the state of each of its channels by the information available in its Communications State Table (CST). In this model, seven active states and one null state exist for each communication channel. These states are identified in Table II.

Table II. Communication Channel States.

State	Description
0	No request, no channel
1	Request from KSOS, awaiting location
2	Location known, awaiting authentication
3	Authentication received, channel open
4	Request from Comm Subnet, awaiting key
5	Key received, awaiting authentication
6	Directory update, awaiting authentication
7	Timeout of channel, cleanup in progress

Figure 8 shows the state transitions allowed by the secure network model. Transitions between states are only permitted when the necessary control information (identifiers, keys, and/or message types) is present.

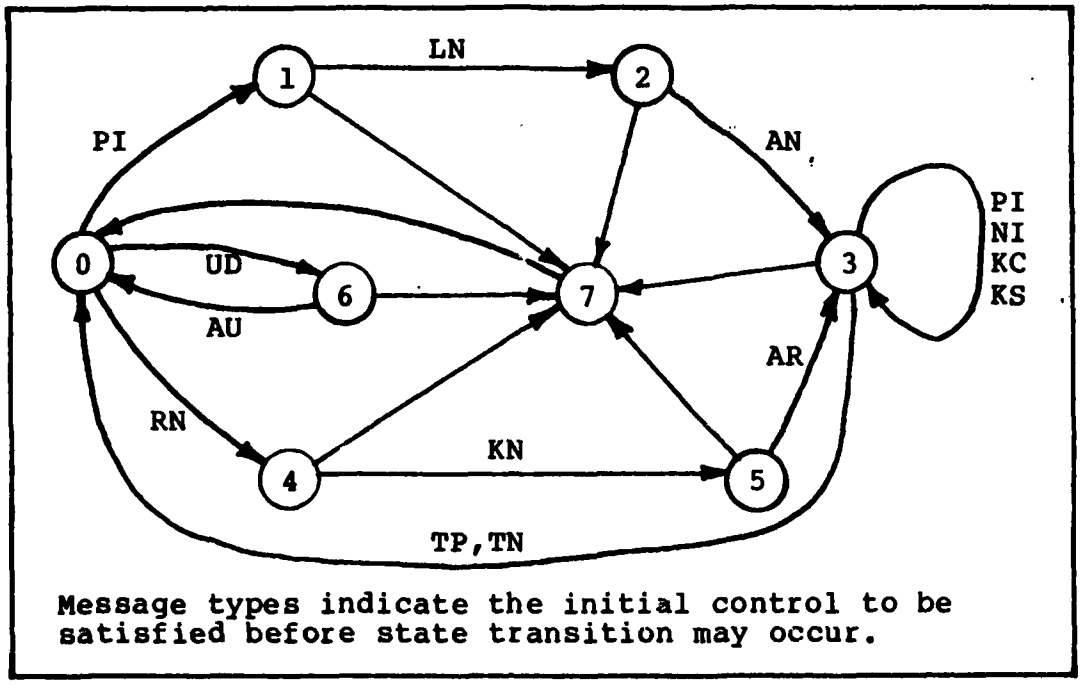


Figure 8. Communication Channel State Transition Diagram.

The initial state for a channel is state 0, the null state, in which no request for the establishment of a secure channel has been made to the SNIC from the KSOS or communications network. All channels are developed from this state and eventually return to this state when the channel is terminated. Theoretically, there exists a channel between every legitimate network subject and every network object. Until a communication is attempted to the object, the theoretical channel is in the null state.

Transition from state 0 to state 1 occurs when the SNIC receives a message from the KSOS which is bound for another network component (message type PI) . Because the operating system of the host is secure, only the operating system is able to pass requests to the SNIC, and since the

communication media connecting the secure operating system of the host to the SNIC is physically secure, the request is always considered legitimate. When the state transition occurs, an initial entry is made in the CST which includes the identifiers supplied by the host for the subject, device, and resource, and a unique identifier generated by the SNIC for this channel. When this is accomplished, an RL message is sent to the NDSC which contains the unique identifier. Because this message is encrypted in the public key of the NDSC, it is assumed that, since the secret key of the NDSC is secure, only the NDSC can correctly interpret the message and is therefore the only other system which knows the unique identifier for this channel.

Transition from state 1 to state 2 is accomplished only if an LN message is received which contains the correct identifier for the channel. The probability of successfully imposing a fraudulent resource location message (type LN), and thereby misdirecting the channel, is dependent on the probability of either guessing the unique identifier for a channel that is in state 1, or successfully changing the directory entry for the location of a resource. Because of the authentication method used in making directory updates, only the system which controls the resource can change its location. Unless the secret key to the controlling system is known, the change cannot be made. Since the secret keys are assumed secure,

the only way to fraudulently impose an erroneous location for a resource is to guess the unique identifier of a channel in state 1. Therefore, the probability of an illegitimate state transition from state 1 to state 2 is the probability of guessing a unique identifier of a channel in state 1. After an LN message is received and decrypted, the CST is updated with the location of the resource, the public key for that location's SNIC, and a new unique identifier for the channel. An RN message is then sent to SNIC serving the resource's location.

Transition from state 2 to state 3 occurs when an AN message is received which contains the unique identifier for the channel. At this point, the key for the channel and the other SNIC's unique channel identifier are entered in the CST. The probability of entering incorrect information is again the probability of guessing a correct unique identifier for a channel in state 2. After this transition is made, an AR message is sent to the other SNIC. The request within this message is encoded in the channel key, thereby establishing another barrier for the security of the information being transferred.

Once a channel reaches state 3 it may only relay four types of messsages. It will accept PI messages received from the KSOS which contain proper subject, device, and resource identifiers; NI messages received from the communications subnetwork which contain the correct unique identifier for the channel and whose information is

encoded in the correct channel key; and KC or KS messages received from the communications subnetwork which contain the correct unique identifier for the channel. When a KC message is received, the channel key is changed in the CST.

It should be noted that the unique identifiers are protected during transmission by their encryption in the public key of the system to which they are sent. If the public keys are only distributed to authorized systems by the NDSC and the distribution of these public keys is protected by their encryption (which is the case in this model), then the possibility of any system other than those validated by the network to obtain the necessary public key to even initiate a message is the same as the probability of decrypting a message without the secret key. Given that the encryption algorithms are sufficiently strong, the probability of decrypting a message without the secret key is essentially nill. Since all of the messages transmitted by the NDSC and the SNIC's are encrypted in the public key of the recipient, and the recipients' secret keys are secure, the messages are secure.

A channel may transition from state 0 to state 4 if an RN message is receivied from the communications subnetwork. Since this message is encoded in the public key of the recipient SNIC, and since only authorized systems are given the public key, the message must either

come from another SNIC or a system which guessed the public key. When the RN message is received, the SNIC creates a channel by making an entry in its CST which includes the subject, device, and resource identifiers, the identifier of the other SNIC, the other SNIC's unique identifier, and its own unique identifier for the channel. It then generates an RK message which contains its unique identifier, encrypts it in the public key of the NDSC, and sends it to the NDSC.

Transition from state 4 to state 5 occurs when a KN message is received which contains the unique identifier sent to the NDSC. Again, the probability of transitioning illegally from state 4 to state 5 is the probability of guessing a unique identifier of a channel in state 4. When the KN message is received, the SNIC generates its own new unique identifier and key for the channel and enters them in the CST. It then transmits an AN message to the other SNIC.

The transition procedures preclude the channel state from transitioning to a secure open channel (state 3) even if an RN message is received from an unauthorized system. Because the receiving SNIC requests the public key of the message originator from the NDSC, prior to responding to the RN message, the response (an AN message) may only be encrypted in the public key of an authorized system (since the NDSC only maintains keys for authorized systems). Therefore, the fraudulent channel being developed as a

result of the fraudulent RN message will be detected when either the NDSC cannot find the requested key (because the system identifier is invalid) or when no correct response is received to the AN message (because the unauthorized system cannot correctly decrypt the identifier in the AN message. In either case, unless a secret key has been compromised, the fraudulent channel will never be completely established and opened.

Transition from state 5 to state 3 occurs when an AR message is received which contains the unique identifier for the channel in state 5. When this transition occurs, the SNIC may decode the the request for the resource with the channel key and pass it through the physically secure channel to the KSOS.

Transition from state 0 to state 6 occurs when a UD message is received from the KSOS. After receipt of a UD message, the SNIC makes an entry in the CST which includes the identifier of the resource to be changes in the directory, the type of change, and a unique identifier for the change. A DU message, containing the unique identifier is then sent to the NDSC.

Transition from state 6 to state 0 occurs when the SNIC receives an AU message from the NDSC which contains the unique identifier for the change. Before the transition is completed, however, an AD mesage is sent to the NDSC to authenticate the change initiated by the original UD message.

Channels may also return to state 0 directly from state 3 upon notification from the KSOS that the subject has terminated his requirement for the resource (message type TP), or upon notification from the other SNIC that the channel has been terminated (message type TN). Any necessary accounting for the channel should be accomplished prior to the return to state 0.

A transition may be made from any active state (1 through 6) to state 7 if no messages traverse the channel within the specified period of time. This prevents the CST from maintaining idle channels and allows the establishment of new channels for waiting messages if some error has prevented the completion of their channels. After any necessary auditing or cleanup, the channel returns to state 0.

During the life of a channel, if any messages are received which do not contain all of the appropriate control information for the current state, an auditing procedure is performed. The specific procedure performed would depend on the irregularity discovered. The model presented does not contain a complete array of auditing procedures, but merely identifies the problem discovered and the message involved. The simulation developed merely records the discrepancy and message. Since the message is removed from the channel by the SNIC, it has no impact on the channel's operation and remains unknown to the processes using the channel. Much more sophisticated

auditing techniques could be incorporated in the network. It should be observed that the auditing procedure may be performed even if the channel concerned is in state 0, since invalid messages may be received for channels which are not being established.

The allowable state transitions and the control information required to accomplish them are shown in Table III. No other transitions are possible in this model.

Table III. Communication Channel State Transitions.

State	Message Type	Other Controls	Next State
0	PI	SDR	1
0	RN	P	4
0	UD	SDR	6
1	LN	P,ID	2
1	--	T	7
2	AN	P,ID	3
3	PI	SDR	3
3	NI	P,ID,K	3
3	KC	P,ID	3
3	KS	P,ID	3
3	TP	SDR	0
3	TN	P,ID	0
3	--	T	7
4	KN	P,ID	5
4	--	T	7
5	AR	P,ID	3
5	--	T	7
6	AU	P,ID	0
6	--	T	7
7	--	--	0

SDR = Subject, Device, and Resource identifiers
P = Message is encrypted in SNIC's public key
ID = Message contains proper unique channel identifier
K = Information is encrypted in proper channel key
T = Timeout of channel

The communications network only establishes secure logical channels between network components. The establishment of a channel, however, in no way binds a resource to fulfill the requests made of it. The resource may conduct its own identification and authentication session with the subject before releasing any information to that subject or before performing any tasks for that subject. This allows the resource to control its own security.

Because the secure logical channels may only be developed and operated under the strict procedures outlined above, there is no way to either divert or subvert them without obtaining the necessary control information or gaining physical access to the secure operating system of the host or the SNIC. Therefore, since the control information is carefully protected during channel development, the only means of inserting fraudulent messages into the network, and having them accepted is to correctly guess the necessary control information. If the probability of correctly guessing the necessary control information is established by setting the length of that information (which thereby controls the number of combinations possible), then the network may be made as secure as desired. In this model, the minimum amount of correct control information required to insert an acceptable message in the network is the public key of the receiving SNIC and a valid channel identifier. It

should be noted that even if a single message may be inserted into the network, this may not be sufficient to divert or subvert a secure logical channel. In most cases, a sequence of control information and message interceptions would be required. However, even if the channel is diverted, there is no guarantee that the information or requests submitted through the communications network will be accepted by the receiving subject.

Because the probability of correctly generating the necessary control information can be determined, and the sequences of acceptable messages necessary to establish a channel from each state are known, the probability of security compromise can be evaluated. The probability of compromise is the highest probability of entering a sequence which will allow a SNIC to accept a message containing actual information for a subject or resource.

Of course, it must be realized that flaws in the physical security of the systems may compromise the security of the information. Therefore, strong physical security measures are necessary to protect the operating systems if the network is to be considered secure.

Design Analysis

The design of the model is conceptually sound, and has been demonstrated in the previous section. There are however several areas which should be discussed further to either eliminate any ambiguity or to identify possible

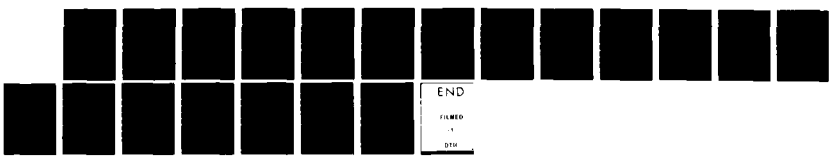
enhancements.

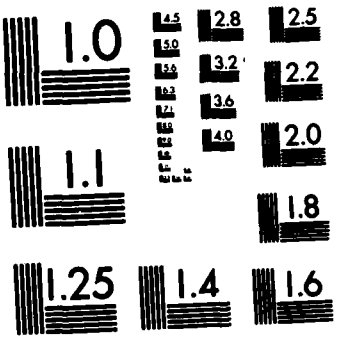
The security of the information is based on the probability of a potential intruder overcoming a series of security barriers. The sequencing of these barriers, which include resource location, identification, and authentication is important. Most of these barriers are essential to the security of the information. The design strengthens some of these barriers, intentionally and unintentionally by the sequencing of these barriers. For example, double encryption of the actual information passed between the SNICs is not necessary, but does provide an additional barrier for security of the information and further restricts the possibility of information compromise. Therefore, a complete analysis of the side effects of contemplated changes should be made prior to making the change.

Layer 6 of the network protocol has not been specified. Layer 6, the presentation layer, is responsible for translating between the KSOS and layer 5 of the SNIC. The design of layer 6 of the network protocol which is implemented in each SNIC must accurately translate the identifiers and message types given by the KSOS to the Standard Network Form. This would be a simple task if the same type operating system is attached to every SNIC, for then very little translation is required. If, however, different types of operating systems are connected in the network, then a separate version of layer

AD-A124 820 A SECURE COMPUTER NETWORK(U) AIR FORCE INST OF TECH 2/2
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
J S STEINMETZ NOV 82 AFIT/GCS/EE/82D-34

UNCLASSIFIED F/G 9/2 NL





6 must be developed for each type of operating system. The translations could be extremely difficult to develop if different operating systems are used. The magnitude of this task should be considered carefully when deciding which operating systems may be interfaced by the communications subnetwork. The development of this translation capability, on the other hand, could allow an enourmous advantage because any secure operating system (for which a translation has been developed) could be used in the network.

The auditing portion of this model is sufficient to notify system managers of difficulties or attempts to subvert the system, but could be improved to actively pursue potential threats to the network. Notifications of audited messages should also be sent to the NDSC, so that a central monitor may audit the network. This would require additional message types to be handled by the NDSC, but would allow a more global observation of the network.

The design and specification of the NDSC purposely left the implementation of the directory vague. The NDSC, however, should be more completely specified. If the NDSC is developed as a specialized SNIC, which is merely an interface to a system which contains the directory, then, even though the NDSC is a unique system in the network, its functions would more closely represent that of an interface computer. This does not suggest that the

directory system not be controlled by the same group that controls the communications network, but merely allows for a more modular implementation. The system containing the directory should be controlled by the managers of the communications network since it provides a significant portion of the security barriers used to secure the network. This system could also be used to maintain auditing information and to monitor the physical security of the individual SNICs. Such tasks would establish the NDSC system as a security center as well as a network directory.

The weakest portion of the network is currently related to the KS message. If a system has the message format, the public key of a SNIC, and a valid channel identifier, it can change the public key used for existing channels which are connected to the same SNIC as the channel with that identifier. This would allow the interception of those messages from the SNIC, whose public key is known, which are bound for the other SNIC controlling the identified channel. It would also permit messages to be sent to the identified channel. A more secure method of changing public keys might be used. The attempt to reduce message traffic by only requiring one KS message to change all of the public keys of the sending SNIC within the receiving SNIC may not be acceptable. Perhaps one message could be sent over each channel, with either some form of verification between the channels

which use that SNIC or some form of verification by the NDSC. Another method which may be used is to prearrange all public key changes through a distribution system outside the network. In any case, further consideration should be given to this area of the model since the potential gain for determining a channel identifier and the public key of the SNIC controlling that channel is greater than any other breakthrough in the communications subnetwork.

There are also problems in the design related to updating the directory when a public key changes. The method for updating the directory is secure, but the time required to make the change may permit some messages encrypted in the old key to be sent to the SNIC which is making the change. These messages would be rejected because they are encrypted in the wrong key. The loss of a required message during channel establishment would prevent the channel from transitioning to a completed state. The channel would be terminated with a timeout condition. Therefore, either the loss of these messages must be accepted or some method of preventing their loss must be established. Loss may be prevented by several methods. The layers may be synchronized and key changes may be transmitted far enough in advance of the actual effective time to permit each system to make the change at exactly the same time. Another method would be to maintain old keys in the SNIC for some period of time

after the change to permit the handling of messages which are encrypted in the old key.

Implementation Considerations

Several factors should be considered when attempting to implement the model presented. Most of these are obvious, but are mentioned for emphasis.

The design was developed with the thought that the SNICs and NDSC would be physically separate from the systems they connect to the secure communications network. This allows for the development and management of separate physical and procedural security measures. The physical security of the communications network may therefore remain independant of the attached computer systems.

The design was also developed with the notion that the actual communication transfer capability of the network (layers one through four) is reliable. Implementation of this network with an unreliable communication medium, in which many messages or packets must be retransmitted, could cause the time delays involved in setting up the secure logical channels to be quite long.

It should also be stated that any attempts to optimize the efficiency should be made in small, incremental steps to prevent the introduction of errors which may compromise the security of the network. This does not imply that optimization is not possible, but

rather emphasizes the fact that, as far as security is concerned, simplicity in design is much more critical than efficiency. Not only is it easier to verify simple designs, but it is also much easier to maintain them. Therefore, improvements of the algorithms for processing efficiency must be made cautiously, to prevent adverse effects on information security.

The specific appproach used to rectify the problems related to public-key updates may cause additional implementation considerations. For example, the advance notification scheme, in which each concerned system is notified of key changes and given an effective time to make the change, may require that the system clocks be synchronized. If synchronization is necessary, the possible compromise situations which exist with improper synchronization must be considered.

The most important implementation consideration, however, is the method by which the design is to be translated into reality. Even though the system, as designed, is secure, if it is not implemented accurately, it may not function securely. To insure that the design is implemented correctly, carefully controlled auditing and quality control procedures must be developed. Personnel security must supplement the quality control to prevent the introduction of either intentional or unintentional flaws in the system which might bypass the system's security controls.

Accreditation and Operation of Network

Once the communications network has been developed and its security verified, the KSOSs may be attached. There should exist, however, some method of verifying or accrediting the systems attached to the communications network. This accreditation may be conducted by the people who control the communications network, or may be conducted by each of the system managers which are connected to the communications network work and desire to interface with the new system. In any case, the network users must be given some measure of the security, identification capability, and management responsibility for the new system. This will permit the users to decide if the information they control may be processed by that system or any of its devices. Without a method for establishing the trustworthiness of a system, the network users would be reluctant to trust anyone else. This accreditation may involve continuous, real-time monitoring of each system, or may simply require periodic, manual evaluations of the systems.

Auditing and accounting procedures should also be incorporated on the network level. As has been previously discussed, the NDSC system would be an excellent location for incorporating these measures. However, each system, or each SNIC may require a certified security officer to monitor the automatic system auditing. This would provide an immediate response capability for potential compromise

situations.

Additionally, a method must be established for implementing updates and repairs to the network. Not only must the updates and repairs be carefully validated, but they must also be implemented and/or inspected by certified personnel. Again, personnel and procedural security measures must be incorporated with the physical security system to prevent the introduction of flaws in the system security.

Summary

The secure computer network model developed in this thesis provides the initial design required for the full development of a secure computer network. The security of the method can be demonstrated and the areas of possible compromise can be identified to attain a probabilistic measure of that security. While the security of the design can be demonstrated, great care is still required to properly implement that design. The physical isolation of the SNICs provides one of the greatest measures of security to the network because it prevents tampering with the system. The simplicity of the SNIC's functions also play a great role in the safety of the information it transmits to other SNICs. Perhaps the most significant attribute of this design, however, is the modularity, because this modularity allows the development of a few basic components which may then be used to develop vast, secure computer networks.

VII Conclusions and Recommendations

Secure computer networks must be developed if computer systems are to be used to process sensitive or personal information. The secure computer network model presented in this thesis has the potential for full scale development and implementation. At the very least, this thesis provides a valuable, methodical discussion of an approach which merges secure data bases, secure operating systems, and secure communications to develop an environment in which information can be safely processed and exchanged by a network of computer systems. The secure communications network is specified so that it may be developed independently and used as the base system in the development of a complete network. It is not constrained by the type of communication media used to actually transmit the information, since the security features are implemented at a higher layer of the protocol. The front-end processing capability of the Secure Network Interface Computer takes the network burdens off of the attached computer system. The modular design of this model also allows for the separation of responsibility in managing and maintaining the network after its development. The security method implemented allows the level of security to be established during the design, rather than determined after system completion. By selecting the length of the keys and identifier to meet

the required probability of information compromise, the system developers are allowed to determine the strength of the security. Not only has this thesis established the method of secure communication, but it also identifies the specific functions required of the system components. These functions are presented in the simple, straightforward manner which is essential for verification and validation. Each feature of this design is a major step toward the development of a complete, secure computer network. The primary intent of this work is, however, to spur the development of secure computer networks. The approach used in this thesis is not the only approach, but is a viable one and does provide many insights about the problems and possibilities of secure network development. This design could provide the foundation of a functional, complete, modular, secure computer network in which the security responsibility for every unit of information is precisely identified.

Conclusions

1. A computer network must be designed to provide a complete, secure environment for the information it processes.

2. Information within a computer network is always in one of three states: storage, processing, or transfer. The information must be secure in each of these states if it is to be considered secure.

3. A complete, secure computer network includes not

only the communications system, but also the operating systems and data bases which use that system. Information security requires the proper interfacing of these systems.

4. The secure communications network must be able to function independantly of any attached operating system if its integrity is to be maintained. This may be accomplished by developing Secure Network Interface Computers (SNICs) which serve as endpoints for the communications subnetwork and perform all of the network communication functions for the attached operating system. These functions include the creation and management of secure logical channels to the desired network resources.

5. Secure logical communication channels may be established between two endpoints by using encryption to protect information carried by unsecure physical channels. The creation process includes locating, identifying, and authenticating the desired resource. Once a secure channel is established, requests, and the responses to those requests may be carried by the channel.

6. The secure communication model presented in Chapter V allows secure communication between subjects residing on separate computer systems which are connected to the same communications network. It also allows for the maintanence of a directory containing the location of every network component, including the information contained in the network.

Recommendations for Further Study

1. The requirements for developing layer 6 of the computer network protocol, which translates information between the attached operating system's form and the Standard Network Form, should be analyzed.

2. The secure communication model should be implemented initially with a network of similar secure operating systems. The requirements for layer 6 would therefore be minimal since all operating systems would use the same form of messages. Implementation of the model would require the construction of a SNIC prototype and an NDSC prototype.

3. The requirements for a directory suitable for locating information within the network should be studied.

4. The SNIC must select messages for processing from its queues in some order. Analysis of the effects of different priority schemes for this selection from the queues should be conducted.

5. A more sophisticated auditing program should be developed.

6. Appropriate encryption and decryption algorithms should be developed or selected for this model.

7. Techniques for identifying individual network users and devices which would reduce the overhead related to identification and authentication should be developed.

8. The secure communications model allows the probability of compromise to be established at a desired

level by adjusting the length of the keys and identifiers. Formal specification of the probaility functions should be made. In addition, an analysis of the physical security measures required for each SNIC and the NDSC to support the probability of compromise established by the secure communications model should be completed.

9. A set of management guidelines and criteria to be used for accreditation of the attached operating systems and maintanence of the communications network should be developed.

10. Analysis of the time delays associated with the various queues and systems should be completed to make it possible to determine the overhead related to the security measures. While the security is necessary, such analysis may allow the overhead to be reduced.

Summary

Secure computer networks must be developed if they are to process sensitive or personal information. The model presented in this thesis is a basis from which these secure networks may be developed. A great deal of work remains to be done before a secure computer network may actually be built and operated. It is imperative, however, for this work to be started if secure computer networks are to become a reality.

Bibliography

Abrams, Marshall D. et al. Tutorial on Computer Security and Integrity. Long Beach, California: IEEE Computer Society, 1977.

Anderson, James P. Computer Technology Planning Study. Fort Washington, Pennsylvania: James P. Anderson & Company, October 1972. (AD 758 206 and AD 772 806).

Andrews, Walter. "Autodin II Dropped -- Too Vunerable," Air Force Times, 42, 40 : 20,23 (April 26,1982).

Berson, T. A. and G. L. Barksdale, Jr. "KSOS - Development Methodology For a Secure Operating System," AFIPS Conference Proceedings, 48: 365-371 (1979 National Computer Conference).

Bochmann, Gregor V. and Tankonano Joachim. "Development and Structure of an X.25 Implementation," IEEE Transactions on Software Engineering, SE-5, 5: 429-439 (September 1979).

Cheheyl, Maureen H. et al. "Verifying Security," Computing Surveys, 13, 3: 279-339 (September 1981).

Davies, Donald W. Tutorial: The Security of Data in Networks. New York, New York: Publishing Services Institute of Electrical and Electronic Engineers, Inc., 1981.

Demillo, Richard A. et al. Foundations of Secure Computation. New York, New York: Academic Press, Inc., 1978.

Denning, Dorothy E. and Peter J. Denning. "Data Security," Computing Surveys, II, 3: 227-249 (September 1979).

Folts, Harold C. "Status Report on New Standards For DTE/DCE Interface Protocols," Computer, 12: 12-19 (September 1979).

Green, Paul E. Jr. Computer Network Architectures and Protocols. New York, New York: Plenum Press, 1982.

Hinckley, A. C. and J. Mitchell. Issues in Computer Network Security. Project No. 527B. Bedford, Massachusetts: The Mitre Corporation, September 1978. (AD A060 007).

Hoffmann, Lance J. Modern Methods for Computer Security and Privacy. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1977.

Hsiao, David K., Douglass S. Kerr, and Stuart E. Madnick. Computer Security. New York, New York: Academic Press, Inc., 1979.

Katzan, Harry J. The Standard Data Encryption Algorthim. New York, New York: Petrocelli Books, Inc., 1977.

Kam, John B. and George I. Davida. "Structured Design of Substitution - Permutation Encryption Networks," IEEE Transactions on Communications, COM-29, 6: 778-786 (June 1981).

Kent, Stephen T. "Security Requirements and Protocols for a Broadcast Scenario," IEEE Transactions on Communications, COM-29, 6: 778-786 (June 1981).

Kline, Charles Steven. Data Security: Operating Systems and Computer Networks. PhD Dissertation. Los Angeles, California: University of California, Los Angeles, 1980. (Contained in AD A103 371).

Konheim, Alan G. "Guest Editor's Prologue," IEEE Transactions on Communications, COM-29, 6: 761 (June 1981).

Kuo, Franklin F. Protocols and Techniques for Data Communication Networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981.

Landwehr, Carl E. "Formal Models for Computer Security," Computing Surveys, 13, 3: 247-278 (September 1981)

Lennon, Richard E., Stephen M. Matyas, and Carl H. Meyer. "Cryptographic Authentication of Time Invariant Quantities," IEEE Transactions on Communications, COM-29, 6: 773-777 (June 1981).

Lientz, Bennet P. and Ira R. Weiss. "On the Evaluation of Reliability and Security Measures in a Computer Network", Tutorial on Computer Security and Integrity. VI-17 - VI-41. Long Beach, California: IEEE Computer Society, 1977.

Lipner, Steven B. "Secure Computer Systems For Network Applications," Tutorial on Computer Security and Integrity. VI-12 - VI-16. Long Beach, California: IEEE Computer Society, 1977.

McCauley, E. J. and P. J. Drongowski. "KSOS - The Design of a Secure Operating System," AFIPS Conference Proceedings, 48: 345-353 (1979 National Computer Conference).

Padlipsky, M. A., D. W. Snow, and P. A. Karger. Limitations of End-to-End Encryption in Secure Computer Networks. Project No. 672B. Bedford, Massachusetts: The Mitre Corporation, August 1978. (AD A059 221).

Padlipsky, M. A., et al. "KSOS - Computer Network Applications," AFIPS Conference Proceedings, 48: 373-381 (1979 National Computer Conference).

Parker, Donn B. Crime by Computer. New York, New York: Charles Scribner & Sons, 1976.

Parker, Donn B. Computer Security Management. Reston, Virginia: Reston Publishing Company, Inc., 1981.

Popek, G. J. and C. S. Kline. "Issues in Kernel Design," AFIPS Conference Proceedings, 47: 1079-1086 (1978 National Computer Conference).

Popek, G. J., et al. "UCLA Secure Unix," AFIPS Conference Proceedings, 48: 355-364 (1979 National Computer Conference).

Popek, Gerald J. and Charles S. Kline. "Encryption and Secure Computer Networks," Computing Surveys, II, 4: 331-356 (December 1979).

Popek, Gerald J. Secure Reliable Processing Systems: Semi-Annual Technical Report, July 1979 - June 1981. Los Angeles, California: Computer Science Department, University of California, Los Angeles, July 1981. (AD A103 371).

Schell, Roger R., Peter J. Downing, and Gerald J. Popek. Preliminary Notes on the Design of Secure Military Computer Systems. Technical Report. Hanscom AFB,Massachusetts: Electronic Systems Division, January 1973. (AD A089 433).

Schacht, J. M. Jobstream Separator System Design. Bedford, Massachusetts: The Mitre Corporation, September 1975. (AD A016 403).

Scherf, J. A. Computer and Data Security: A Comprehensive and Annotated Bibliography. Cambridge Project MAC. Cambridge, Massachusetts: Massachusetts Institute of Technology, January 1974.

Shen, J. T. Multilevel Security for Computer System Networks: A Survey and Discussion. San Diego, California: Naval Electronics Laboratory Center, 7 May 1974. (AD 919 838).

Smid, Miles E. "Integrating the Data Encryption Standard Into Computer Networks," IEEE Transactions on Communications, COM-29, 6: 762-772 (June 1981).

Sunshine, Carl A. "Formal Techniques for Protocol Specification and Verification," Computer, 12: 20-27 (September 1979).

Sunshine, Carl A. Communication Protocol Modeling. Dedham, Massachusetts: Artech House, Inc., 1981.

Tanenbaum, Andrew S. "Network Protocols," Computing Surveys, 13, 4: 453-489 (December 1981).

Tasker, P. S. and D. E. Bell. Design and Certification Approach: Secure Communications Processor. Bedford, Massachusetts: The Mitre Corporation, June 1973. (AD 765 518).

Turn, Rein and Willis H. Ware. "Privacy and Security Issues in Information Systems," IEEE Transactions on Computers, C-25, 12: 1353-1361 (December 1976).

Turn, Rein. Advances in Computer System Security. Dedham, Massachusetts: Artech House, Inc., 1981.

Walker, Bruce J. and Ian F. Blake. Computer Security and Protection Structures. Stroudsburg, Pennsylvania: Dowden, Hutchinson, & Ross, Inc., 1977.

Ware, Willis H. Security Controls for Computer Systems. Report of Defense Science Board Task Force on Computer Security. Washington, D. C.: Office of the Secretary of Defense, Revised October 1979. (AD A076 617).

Winkler, Stanley and Lee Danner. "Data Security in the Computer Communication Environment," Computer, 7: 23-31 (February 1974).

Zimmermann, H. "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," IEEE Transactions on Communication, COM-28: 425-432 (April 1980).

VITA

Jay Stephen Steinmetz was born on 16 May 1954 in South Charleston, West Virginia. He graduated from high school in Ponce, Puerto Rico in 1972 and attended the United States Air Force Academy, where he majored in Computer Science and received the degree of Bachelor of Science in June 1976. Upon graduation he attended USAF pilot training and received an areonautical rating in June 1977. He then served as a C-141 pilot and aircraft commander in the 14th Military Airlift Squadron, 63rd Military Airlift Wing, Norton AFB, California until entering the School of Engineering, Air Force Institute of Technology, in June 1981. He is a member of Tau Beta Pi.

Permanent Address: 4310 Long Grove Drive
Seabrook, Texas 77586

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/EE/82D-34	2. GOVT ACCESSION NO. A124 820	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A SECURE COMPUTER NETWORK		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jay S. Steinmetz Capt USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 113
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for Public release: IAW AFR 190-17.

LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH

4 JAN 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Networks
Computer Information Security
Secure Communications

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this thesis, the initial design for a secure computer network is developed. The requirement for a secure computer network is based on the need to protect sensitive and personal information currently processed by computer networks. The concepts of physical security, reference monitors, encryption, and network protocols are presented. Then, the top-level design of the secure computer network is developed. This design consists of secure data bases controlled by kernelized secure operating systems which are connected by a secure communications network. The phases of secure communications: location,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20.
identification, request, and request response are discussed. A model for the secure communications network is then presented. This model relies on two major components: Secure Network Interface Computers (SNICs) and a Network Directory and Security Center (NDSC). A finite state analysis of the communications channels demonstrates the security of the model. Recommendations are presented to continue the development of this secure network.

END

FILMED

3-83

DTIC